STANFORD ARTIFICIAL INTELLIGENCE PROJECT
MEMO AIM-143

COMPUTER SCIENCE DEPARTMENT
REPORT NO. CS-209

PROJECT TECHNICAL REPORT

BY

John McCarthy, Arthur Samuel, and the Artificial Intelligence Project
Edward Feigenbaum, Joshua Lederberg and the
Heuristic Programming Project Staff

MARCH 1971

ARPA Order No. 457

COMPUTER SCIENCE DEPARTMENT
STANFORD UNIVERSITY

82

STANFORD ARTIFICIAL INTELLIGENCE PROJECT MARCH 1971
MEMO AIM-143

Project Technical Report

by

John McCarthy, Arthur Samuel, and the Artificial Intelligence Project
Edward Feigenbaum, Joshua Lederberg and the Heuristic Programming Project Staff.

ABSTRACT: An overview is presented of current research at Stanford in artificial intelligence and heuristic programming. This report is largely the text of a proposal to the Advanced Research Projects Agency for fiscal years 1972-3.

The research reported here was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense under Contract SD-183 and in part by the National Institutes of Mental Health under Grant PHS MH Ø66-45-Ø8.

Reproduced in the USA. Available from the Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia 22151. Price: full size copy, $3.ØØ; microfiche copy, $.95.

Table of Contents

1. Artificial Intelligence Project

Artificial Intelligence is the experimental and theoretical study of perceptual and intellectual processes using computers. Its ultimate goal is to understand these processes well enough to make a computer perceive, understand and act in ways now only possible for humans. This understanding is at present in a very preliminary state. Nevertheless, progress in identifying and duplicating intellectual mechanisms is being made and the range of problems that computers can be made to solve is increasing. The understanding so far achieved has important potential practical applications. The development of these applications is worth undertaking.

The Stanford Artificial Intelligence Project is concerned with both the central problems of artificial intelligence and some related subfields of computer science. The proposed structure of the Project is given in Figure 1. The scopes of some continuing activities have been modified and two new research areas have been added: Analysis of Algorithms and Machine Translation.

Figure 1. Structure of the Stanford Artificial Intelligence Project

```
 --------------------------------------------------------------------------------
 |          |                |           |            |               |            |
 |          |                |           |            |               |            |
 |   -------|--------        |   --------|-------     |       --------|------      |
 |  | Interaction  |         |  | Heuristics    |     |      | Mathematical |       |
 |  |   with the   |         |  |               |     |      |  Theory of   |       |
 |  |Physical World|         |  |               |     |      | Computation  |       |
 |  |______________|         |  |_______________|     |      |______________|       |
 |   Binford,Feldman         |   Luckham,Samuel       |       Floyd, Knuth         |
 |   Kay, Samuel             |                        |       Manna, McCarthy      |
 |                           |                        |                            |
 |                           |                        |                            |
 -----------            -----|-----          ---------|--------          -----------
|Analysis  |           |Mechanical |        |Representation  |        | Models of  |
|   of     |           |Translation|        |    Theory      |        | Cognitive  |
|Algorithms|           |           |        |                |        | Processes  |
|__________|           |___________|        |________________|        |____________|
 Knuth                  Schank,Wilks         McCarthy                  Colby
                        McCarthy
```

"Analysis of Algorithms" is headed by Professor Donald Knuth and directed to an understanding of the quantitative behavior of particular algorithms. The properties of many algorithms that are of central importance to computer science are known only in a qualitative or crudely quantitative way. Knuth and his group are employing analytical techniques to deepen our knowledge of this area.

The problem of machine translation will be approached anew from two directions: artificial intelligence and linguistics. This small project will involve representatives of both disciplines who propose to test their ideas initially on a restricted formal language.

"Interaction with the Physical World" includes continuing projects on computer vision and control, as well as speech recognition research. During Prof. Feldman's sabbatical leave, (academic year 1970-71) responsibility for Hand-Eye research has passed to Drs. Thomas Binford and Alan Kay. Work on speech recognition was curtailed with the departure of Professor Reddy but is continuing in the area of syntax-directed recognition.

Work on Heuristics continues in the areas of machine learning and automatic deduction. Board games such as Checkers and Go are the primary test vehicles for ideas in machine learning. Theorem-proving is the current objective of our research on automatic deduction.

John McCarthy's Representation Theory work will continue on epistemological problems (i.e. choosing a suitable representation for situations and the rules that describe how situations change).

Research in Mathematical Theory of Computation is expanding somewhat, partly through other sources of support. A practical goal of this work is to replace certain time-consuming and uncertain program debugging processes with formal proofs of the correctness of programs.

The work on Models of Cognitive Processes shown in Figure 1 is an affiliated project not included in this proposal. It will be supported by the National Institutes of Mental Health under Grant MH06645-10.

Subsequent sections cover the proposed research in somewhat more detail.

1.1 ANALYSIS OF ALGORITHMS (Donald Knuth)

"Analysis of algorithms" is a field of study directed to an understanding of the behavior of particular algorithms. Two kinds of problems are usually investigated:

A. Quantitative analysis of an algorithm. In this case the goal is usually to determine the running time and/or memory space requirements of a given algorithm. The determination of running time can be done in an essentially machine-independent manner by expressing the algorithm in some machine- independent language (not necessarily a formal language) and counting the number of times each step is executed. Usually these counts include a "worst case analysis" (the maximum number of times that the step can be performed, taken over some specified set of inputs to the algorithm); a "best case analysis" (the minimum number of times), and a "typical case analysis" (the average number of times for a given distribution of inputs). It is in fact desirable to have complete information about the distribution of the number of times, for a known distribution of the input, whenever this can be worked out. A typical example of such an analysis is presented in detail in [1, pp. 95-99].

B. Determination of "optimal" algorithms. In this case the goal is usually to find the "best possible" algorithm in a given class of algorithms. We set up some definition of "best possible" which reflects, as realistically as possible, the pertinent characteristics of the hardware which is to be associated with the algorithm. A typical example of this sort of analysis, applied to the problem of computing x^n with the fewest multiplications when n is fixed, is discussed in detail in [2, pp. 401-418].

Analyses of type A are usually employed when comparing two different algorithms that do the same job, to see which is more suitable on a particular computer for some particular type of input data. Since there is usually more than one way to solve a problem, analyses of this type can be very helpful in deciding which of several algorithms should be chosen. Occasionally type A analyses are also incorporated into the algorithms themselves; for example, the "spectral test" algorithm [2, pp. 93-96] carries out one type of iteration until it finds that the data has been transformed enough to let another type of iteration complete the job in a reasonable amount of time.

It may seem that type B analyses are far superior to type A analyses; we will have found the "best" algorithm once and for all, instead of performing type A analyses of all algorithms in the class. But this is only true to a limited extent, since slight changes in the definition of "best possible" can significantly affect which algorithm is best. For example, x^{31} cannot be calculated (starting with the value of x) in fewer than 9 multiplications, but it can be evaluated with only 6 arithmetic operations if division is allowed. Klyuyev and Kokovkin-Scherbak [3] proved that the Gaussian

elimination method for matrix inversion uses the minimum number of arithmetic operations, provided that whole rows are always operated on at a time; but Strassen [4] has recently discovered that substantially fewer operations are needed if the row restriction is dropped.

Another problem with type B analyses is that, even when a simple definition of "best possible" is postulated, the determination of an optimal algorithm is exceedingly difficult. For example, the following basic problems are among those not yet completely resolved:

(a) The minimum number of multiplications to compute x↑n given x with n fixed.

(b) The minimum number of arithmetic operations to compute a general polynomial a(n)x↑n + ... + a(1) + a(0), given x, for fixed values of the coefficients.

(c) The minimum number of steps needed to multiply two given binary n-bit numbers.

(d) The minimum number of steps needed to recognize whether a given string belongs to a given context-free language.

(e) The minimum number of steps needed to multiply two given n x n matrices, when n is known.

(f) The minimum number of comparison steps needed to sort n elements.

Asymptotic solutions are known for problems (a) and (f), and the solution to (b) is known within 1 or 2 operations, for "almost all" polynomials; but in cases (c), (d), and (e) the known upper and lower bounds for the desired quantities are far apart. No asymptotic solution to problem (f) is known when simultaneous comparisons are allowed. The evidence in case (a) suggest strongly that the exact answer as a function of n has no simple form which will ever be discovered without exhaustive trial-and-error search. Furthermore, the simplified definitions of "best possible" often fail to represent sufficiently realistic situations; for example, items need not be sorted by means of comparisons at all, they can be sorted by using bit inspection or by using identities like

min(a,b) = (a + b - |b-a|)/2

If only the number of comparisons is considered, other important characteristics of the sorting problem (e.g., the logical complexity of the program and of the data structures) are ignored. Therefore although type B analyses are extremely interesting, type A analyses more often pay off in practice.

It is obvious that algorithmic analysis is desirable, but is it really a unified discipline? If each algorithm presented a problem quite different from each other algorithm, Analysis of Algorithms would be no more than a hodgepodge of isolated techniques, not deserving any distinction as a special branch of Computer Science. Fortunately, experience has shown that a good deal of unity is present in this area, with the same techniques applied repeatedly. Some of the most important unifying principles are "Kirchhoff's" conservation law for flowcharts; the use of generating functions; discrete calculus; and some aspects of information theory and of automata theory. (For numerous illustrations of these techniques, see the listings under "Analysis of algorithms" in the index to [1] and to [2]). Furthermore, it is not uncommon to find that the analysis of one algorithm applies perfectly to the analysis of another superficially unrelated algorithm. (For example, "radix-exchange sorting" is governed by essentially the same laws as a certain form of "trie memory".) The analysis of an algorithm to find the maximum of a set involves the same formula as does the analysis of the number of cycles in a random permutation, and the running time of this algorithm is strongly related to the storage space required by the "reservoir sampling" algorithm. Algorithmic analysis is coming to be a coherent discipline.

By any definition of Computer Science, the field of algorithmic analysis probably lies in the central core of the subject; it is somewhat surprising that so litte concentrated study has been devoted to it. It may be argued that we shouldn't spend too much time analyzing our algorithms, lest we never get anything else done. This is quite true up to a point, since it is certainly unnecessary for a person to analyze carefully every program that he writes; but there are many algorithms which have special importance because they are applied to so many different problems. These algorithms must be well understoood if Computer Science is to advance significantly.

REFERENCES

[1] Donald E. Knuth, The Art of Computer Programming, vol.1 (Addison-Wesley, 1968).

[2] Donald E. Knuth, The Art of Computer Programming, vol.2 (Addison-Wesley, 1969).

[3] V.V. Klyuyev and N.I. Kokovkin-Shcherbak, "On the minimization of the number of arithmetic operations for the solution of linear algebraic systems of equations", translated by G.J. Tee, Tech. Report CS 24 (Stanford: Computer Science Department, June 14, 1965).

[4] Volker Strassen, "Gaussian elimination is not optimal", Num. Math 13 (1969), pp. 354-356.

[5] Niklaus Wirth, "Closing word", Tenth Anniversary ALGOL Colloquim, Zurich, May 31, 1968 [ALGOL Bulletin 20, p. 16].

1.2 Mechanical Translation (J. McCarthy, R. Schank, Y. Wilks)

We plan to undertake a small effort in mechanical translation. The first effort will be to create programs to translate from a restricted formal language RFL into both English and French. The idea is that RFL will be used to express the semantic content of the sentences independent of grammar and without the syntactic and semantic ambiguities. There are two views of semantic content of natural language held in the project and elsewhere, and both will be explored, probably to the extent of making two translators.

The first view (the linguists in the project are betting on this one) is that semantic content (at least to the extent necessary for translation) can adequately be expressed by some form of non-logical representation, such as a network structure.

The second view (held by AI people like McCarthy and Sandewall) is that the semantics of natural language will have to be developed along lines similar to those taken in mathematical logic, i.e. the notion of denotation for phrases and sentences of natural language will have to be formalized. From this point of view, the first cut at RFL should be based on the predicate calculus, and a major effort should go into devising predicates that will enable the content of a wide class of sentences of natural languages to be expressed.

From the linguistic end, Yorick Wilks and Roger Schank will lead the work aided 1/4 time by Dr. A.F. Parker-Rhodes and with Dr. Margaret Masterman as a consultant. From the AI direction, McCarthy and perhaps Patrick Hayes will take part. Several research assistants will also be involved.

One major reason for reopening the mechanical translation problem is that considerable advances have been made towards the satisfactory semantic representation of natural language material. Another is that there has been an increase in the general level of sophistication about the delicacy with which real natural language forms must be treated.

MT went through a certain number of clearly definable phases. There was the word-for-word translation stage, at the inception of MT. When that failed, linguists were called on to remedy the situation. Their emphasis on syntactic structure caused a shift towards more formal methods in MT, most of them based on the work of Chomsky. When that approach failed, it became increasingly clear that something called semantics ought to be added, but no one was quite sure what that was. Workers in MT began to fall into three groups: those who thought that a great deal more lexicography was needed; those who felt that the lack of an adequate theory of human language understanding and generation must be dealt with before more effort on MT could be expended; and those who were so heavily into their own approaches that they failed to see troubles because of their belief that the goal was just over the next mountain.

The ALPAC Report caused most government financing to be withdrawn from MT, largely because it was felt that MT was too expensive as human translators could do the job more cheaply. Since the ALPAC Report [1966], researchers have made considerable strides in the development of a theory of natural language understanding (for discussion of this, see J. Mey [1970]). The impetus for these has been provided by the growth of time-shared computer systems that permit on-line dialogues between man and machine. The desire to use natural language as the medium of communication in these projects has necessitated the development of a sufficiently rich formal structure that can represent the conceptual content of each natural language sentence typed into the machine. Most importantly, these formal structures have been built to handle linguistic input at a higher level than that of sentences, and in conjunction with memory of earlier input, and the long-term memory of the computer model. One aim of these interlingual formal models has been that inferences, logical operations and implications may work in conjunction with the analyzed content of an utterance so as to establish the intent of the utterance and its affect on the memory. It was found by a number of researchers that formal interlingual structures could be made to direct language analysis so as to eliminate previous reliance on syntactic analysis, and replace it with a more heuristic approach to sentence structure.

We propose using some of the approaches that have recently been developed for dealing with computational linguistic and formal linguistic and formal logical problems in order to enrich the emerging theory of human language understanding and generation, and to apply these theoretical and practical advances to the problem of translating languages by computer. For example, we would combine the approaches of the following: (1) John McCarthy's suggestions for the construction of a logic-based intermediate language between source language (S) and target language (T), (2) Schank's system of representing semantic structure based on an interlingual vocabulary and a system of dependency relations, (3) the English-French MT work of Shillan and Rutherford based on a theory of phrase-for-phrase translation, (4) Wilks' system of semantic structure representation using a mixture of dependency and phrase-structure rules ranging over semantic objects. A number of parallels and contrasts between these approaches could be explored, but what is most important is that they all share certain assumptions, namely that conventional grammatical analysis is not fundamental, whereas some intermediate and interlingual representation between S and T is, so then it is proposed to make a new attack on the MT problem. Using an interlingua to detect and transform semantic content, thus the approach will be, from the start, in principle interlingual though the initial system set up will be between two languages only-- English and French.

Interlinguas for MT can be of two kinds:

(A) Axiomatized formal systems with separate rules to fit the formulas generated by the system to some given natural language. The early Masterman-Parker Rhodes semantic product-lattice system was of this intermediate "Progression Calculus" to connect sub-lattices of the deep semantic system to segments of surface structure.

It cannot be decided a priori that to solve the natural fit problem realistically with such calculi is impossible. Nonetheless, after 15 years of trial and error, the Cambridge Language Research Unit decided to implement, in a contract for the Canadian government, a system with an interlingua of type (B) below initially, in order to gain more knowledge of how to operate, in actual practice, an interlingua of type (A).

(B) General machine algorithms for defining, structuring and interconnecting segments of semantically coded text in any language. At least two variants of such systems have been tried: the first takes the whole sentence, dominated by the verb, as the unit of semantic content. If this approach could be sufficiently simplified it would probably be the shortest and best way to MT. Schank and his students have been making significant strides in this direction. When making such codings for Richens' system the CLRU rate was about one coding per session. His system had a hierarchy of subscripts, whereas the Schank system has a much easier to read hierarchy of arrows.

The second variant of this type relies on an initial segmentation of input sentences into phrasings of the order of length of a single clause or phrase, and intended to correspond to a breath group of a human speaker. The dictionary making required for such an approach can be simpler and more rapid since the paradigm is much shorter. This approach is used in the work of Shillan and Rutherford (3 above) and that of Wilks (4 above) which is still, according to Simmons' survey, the only semantic work working beyond the boundary of the sentence.

Wilks' work has tried to tackle another question that comes up in connection with an interlingua: namely, how large are the content patterns that can be transferred from one language to another. This work has been on the semantic discourse structure of paragraphs and has imposed semantic patterns, or templates, onto the segmented and coded text. Its most general assumption has been that it is the bigger patterns in language, applying over the longest available text length, that are genuinely interlingual: and, moreover, that interlinguality at that level can to some extent compensate for the obvious failures of interlinguality at the bottom or coding level, where the actual linguistic codings used are inevitably biased towards either the S or T language.

It should be possible to combine the approaches developed at the Cambridge Language Research Unit with those developed at Stanford so as to integrate a system that utilizes semantic networks to represent

the interlingual content of a piece of discourse. The possibility of utilizing networks by conjunction with certain associative criteria and properties of a general memory structure to establish the meaning of the speaker as opposed to the meaning of the sentence is currently being investigated by Schank. Presumably the entire procedure can be combined with certain high level logical operations in order to create a final representation that could serve as the starting point of a generation routine for natural language. A major goal of this project, therefore, for those mathematical- interlingua oriented, but who have become practically wary, is to develop interlingual systems that are formally and algorithmically interesting, yet which can have natural language dictionaries made for them in a simple and straightforward manner. The stress should not be on contrast and comparison between the constituent approaches to this proposed project, but on the degree to which the different approaches complement one another, and supply elements missing in the others.

REFERENCES

[1] ALPAC Report, "Language and Machines", U.S. Government Printing Office, Washington D.C., 1966.

[2] M. Masterman, "Semantic Interlinguas for Message Detection", Final report on ONR Contract, Cambridge Language Research Unit, 1970.

[3] J. McCarthy and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", in Machine Intelligence 4, Edinburgh. 1969.

[4] R. Montague, "English as a Formal Language", Linguaggi nella Societa e nella Technica, Milano, 1970.

[5] R. Schank, L.Tesler, and S. Weber,"Conceptual Case-Based Natural Language Analysis", Stanford Artificial Intelligence Project Memo AIM-109, 1970.

[6] R. Simmons, "Natural Language Question Answering Systems", University of Texas, 1969.

[7] Y. Wilks,"On-Line Semantic Analysis of English Texts",Journal of Mechanical Translation and Computational Linguistics, December 1968.

[8] J. Mey, "Towards a Theory of Computational Linguistics", paper given at 1970 ACL, Ohio State.

1.3 INTERACTION WITH THE PHYSICAL WORLD

Computer vision in the three-dimensional world and manipulation with mechanical arms continues to be a central interest of the project. Speech recognition reseach continues at a modest level.

1.3.1 Hand-Eye Research (Thomas Binford, Jerome Feldman, Alan Kay).

In our research we aim to develop the ability to see and manipulate in simple industrial situations. These simple problems are beyond our abilities now, but we expect modest gains in the next two years which will stimulate use of newly available hardware. Mechanical arms now have a place in industry; touch sensing and computer control have potential advantages in extending versatility. We anticipate the time when it will be simpler and cheaper to use a general purpose device than to make special purpose machines, just as computers have replaced many special purpose control devices.

Visual perception solutions have applications in advancing the capabilities of computer systems, by making easier communication with computers, and as visualization tools in problem solving and natural language.

Arm and eye modules are sufficiently standard that a handful of people use them routinely. The Hand/Eye submonitor [1] coordinates about 7 cooperating jobs (there will be 3 or 4 more soon) which communicate by message procedures and a global model in the common upper segment. An ALGOL style language SAIL, implemented by Robert Sproull and Dan Swinehart [2,3] is in general use by the Hand/Eye group and others at the project.

A user package for the arm has made it widely available, and generalized manipulation procedures and improved solutions have stimulated extensive use of the arm. A new arm has been constructed and operated [4], a dynamic trajectory servo [5] using a Newtonian mechanics model has operated the new arm. Non-linear calibration, a user package for the new arm, and obstacle avoidance will be done in the first half of 1971.

The visual system is distinguished by automatic sensor accommodation as an integral part of the recognition process [5]. An edge follower has been written which uses automatic accommodation to enhance its dynamic range and selectivity [7]. Camera calibration [8] converts pan and tilt of the camera to space angles and positions on the support plane. Color identification routines have been written by Tenenbaum [6] and Binford. The SIMPLE body recognizer [9] identifies isolated objects from their outlines. The COMPLEX body recognizer [9] is designed for incomplete edge information and incorporates prediction and verification techniques for missing edges. G. Grape [10] is developing an improved program for organization of line drawings from raw edge data, using edge prediction and verification.

In order to develop and test the system and to standardize the independent modules, the task of "Instant Insanity", a puzzle involving colored cubes, was successfully carried out. The strategy job was the work of Bob Sproull, Jerry Feldman and Alan Kay.

We propose to work toward understanding of more complex objects in more complex scenes, such as tools on a workbench or outdoor scenes. Many scenes will be too complex. Over the course of two years we will use new visual functions, organize the scene in new levels, and use a repesentaion of complex objects; we will incorporate these modules in a goal-oriented system which coordinates them.

We will make a region finder which uses color. The new function color permits a new level of organization: regions of homogeneous brightness in three colors will be grouped into super-regions of uniform color. We will benefit from the use of color even with a primitive color perception. What is needed is a fundamental understanding of color perception, color constancy. We have none in sight.

We will organize color super-regions into higher super-regions by proximity in space and color. Color super-regions are defined by connectivity: the set of points with a nearest neighbor having the same property. In very simple scenes of objects with uniform faces, region-oriented or edge-oriented processors are adequate. But when we look out a window, we see sky, trees and grass. In none of these areas will regions based on contingulty enable us to describe that area as a unit. We see patches of fresh green among brown clay or old grass. Patches of sky show through the trees. Patches of blue in the sky are separated by clouds. If we group together the color super-regions into higher super-regions based on proximity in position and color, i.e., we group regions which are nearby but not connected, we describe that outdoor scene in terms of a few main parts: clouds, sky, grass, earth, trees, some of which overlap. We have added another level of organization that in some cases can make sense of what appeared a jumble.

We will criticize the super-regions according to how they simplify the scene. We will find relations within the super-regions and relations among the super-regions. We require many levels of organization. Grouping based on some similarity; sub-grouping based on differences. Group into super-regions based on proximity in space and some attribute (or vector of attributes).

We will form super-regions based on proximity in space and shape, size and directionality. We suggest that we can organize simple textures. The natural language description of texture seems a reasonable representation: the set of texture elements and the geometric relations among them. As texture elements we can work with lines and blobs (whose shape we can describe, perhaps in a primitive way now). We seek to isolate repeated elements. Relations among repeated elements will be examined. This will be a multi-level

process in which the finding of some relations will help in establishing others. Dealing with profiles of linear textures, a primitive program with that structure was able to describe the textures [Wolfe and Binford]. There are three basic operations involved: finding spatial features, two-dimensional shape description, and organization of features. We can do each operation well enough on simple cases to make progress by combining them.

If we were to organize by proximity using usual "clustering" methods, the cost would be prohibitive. These methods rely on computing distances from an element to all other elements. Even if we were to factor the problem into a two-dimensional problem plus a search (search for a match among regions nearby in space, or search among like colors for regions nearby in space), the cost would be prohibitive. A technical aid to organization is the use of proximity in n-space. This can be implemented at reasonable cost in computation and storage by the technique of multi- entry coding [Binford]. In the example of color super-regions, we implied proximity in a four-dimensional space (two color dimensions and two spatial dimensions). In using other attributes we work with similar high-dimensional spaces. This is a reasonable extension of the region-oriented structure in that the cost is slight when the scene is adequately described by region-growing or edge-finding techniques, for then there are few regions and little effort is involved in proximity among the regions. The cases where higher level organization costs something are those cases which could not be handled before.

We warn against the illusion that the visual problem will be solved by one technique. For each new facility we will have "counter-examples", problems it cannot overcome. But we greatly increase the set of problems which are routine for the system which includes that facility.

We will form representations of complex objects. McCarthy [12] has emphasized representation theory as primary in problem-solving. We require a good representation as a compact heuristic base, a source for heuristics which prevents random accumulation and mutual antagonism. We choose representations which are three-dimensional and obtained by the operations of cutting and joining primitive three-dimensional forms.

We will use stereo correlation to obtain motion parallax and foreground/ background separation in stereo images. As emphasized by McCarthy, this is a method of overall characterization of the scene, separating the scene into potentially significant areas. There are some simple cases:

(a) with camera motion or small angle stereo, all disparities are small;
(b) the distant parts of the scene have small disparity;
(c) disparity of ground or floor can be approximately predicted, traced by continuity.

Color and other properties (for example, the "energy", a measure of contrast) narrow the range of search. A good starting place for many objects is the assumption that they rest on the ground.

We will use depth information in the shape representation. Stereo correlation provides a depth mapping. We can use the shape representation to build up a model of the object. We will also use the representation to organize data from a direct-ranging experiment by triangulation.

We will use visual feedback in a variety of ways to control the arm. Immediately, we will control stacking blocks and putting a square pin through a square hole.

As our visual facilities become stronger, we will use visual feedback in tracking the hand, screwing a bolt into a nut, and picking up blob- type objects.

REFERENCES

[1] R. Sproull and K.K. Pingle, in preparation.

[2] D. Swinehart and R. Sproull, SAIL, Operating Note No. 57.1, Stanford Artificial Intelligence Project. (Soon to be superceded by an updated version).

[3] J. Feldman and P.D. Rovner, "An Algol-based Associative Language", Comm. ACM, August 1969.

[4] V. Scheinman, "Design of a Computer Controlled Manipulator", AIM-92, Stanford Artificial Intelligence Project.

[5] R. Paul, in preparation.

[6] J.M. Tenenbaum, "Accommodation in Computer Vision", Thesis Electrical Engineering, Stanford University, November 1970.

[7] J.M. Tenenbaum and K.K. Pingle, in preparation.

[8] I. Sobel, "Camera Models and Machine Perception", AIM-121, Stanford Artificial Intelligence Project.

[9] G. Falk, "Computer Interpretation of Imperfect Line Data as a Three Dimensional Scene", Thesis, Computer Science Department, Stanford University, July 1970.

[10] G. Grape, unpublished.

[11] Wolfe, P.D., Binford, T.O., "Some Methods for Analysis and Description of Linear Visual Texture", (unpublished).

[12] McCarthy, J., Hayes, P., "Some Philosophical Problems from the Standpoint of Artificial Intelligence", Machine Intelligence 4, (D. Michie, ed.), Edinburgh University Press, 1969.

[13] Binford, T.O., "Proximity in N-Dimensions", (in preparation).

1.3.2 Speech Recognition (Arthur Samuel)

Work on speech recognition in the Stanford A.I. Project was slowed by the departure of Prof. Reddy and by the gradual completion of projects being carried on by his students. Studies by Dr. George White (2) and by Dr. Mort Astrahan (1) have, however, pointed to a slightly new direction that our speech work can take. Impetus has been given to this work by recent extensions of Dr. Astrahan's work by Ken Sieberz and by some new ideas which this work has prompted.

Gary Goodman is continuing his thesis research on a syntax-controlled speech recognition system. It is anticipated that several first year graduate students will undertake studies related to speech recognition.

REFERENCES

[1] Astrahan, M.M., "Speech Analysys by Clustering, or the Hyperphoneme Method, AIM-124, Stanford A.I. Project, May 1970.

[2] White, G., "Machine Learning Through Signature Trees, Application to Human Speech", AIM-136, Stanford A.I. Project, October 1970.

1.4 HEURISTICS

1.4.1 Machine Learning (Arthur Samuel)

Work is continuing on the checker program with continued help from K.D. Hanson. A major reprogramming effort has just been completed to reduce the core and disk storage requirements of the program which had gotten completely out of hand. The playing program has been speeded up and it now uses 35K of core as compared with earlier requirements of 55K. The disk requirements have been reduced by rather more than a factor of 2. At the same time the book game storage techniques have been altered to permit the storage of end game situations which could not previously be saved. Mr. Hanson has nearly completed the insertion and editing of 3 books of end game play to be used by the learning program. Several rather drastic modifications of the program are now under consideration. Since each of these would require a major programming effort, an attempt is being made to evaluate the potential usefulness of the different schemes before actually starting the programming.

The Go program now plays a better game than the beginning human player. Work is continuing on development of an evaluation function capable of handling all stages of the game. Difficulties arise from the inherent complexity of Go and from the vast differences in objectives at different times during the game. A straightforward pattern-recognition learning scheme is expected to be working soon to help the program in exact placement of stones. Considerable success has been observed from substituting a local lookahead technique for the more usual generation and search of a (global) move tree. The local lookahead is initiated at points on the board which seem to be critical to the position as a whole, so this technique can be considered a special-purpose pruning heuristic for trees with very large branching factor.

1.4.2 Automatic Deduction (David C. Luckham)

Recently, the interactive resolution theorem-proving program has been extended and improved. The program and its applications to basic axiomatic mathematics is discussed in [1, 2, 3]. Improvements, for example, in the algorithm for the replacement rule for equality (Paramodulation [4]) have led to further successful experiments in which dependencies between axioms in Marshall Hall's third axiomatization of Group Theory [5, p. 6] have been found. The program is being used to search for proofs of theorems of current research interest in several different axiomatic theories. Other improvements include (i) the extension of the input language to many-sorted logic, and (ii) the facility for using natural models (by this we mean the sort of relational structures that occur in everyday use, e.g. a multiplication table for a finite group, or an arrangement of objects in a room) in the model relative deduction strategy. In addition to experiments in theorem-proving, the program

is currently being incorporated into some computer-aided instruction programs at the Stanford Institute for Mathematical Studies in the Social Sciences. The objective here is to increase the flexibility of the instructional programs for high school mathematics by extending the logical inference system to which the student must conform.

One of the obvious weaknesses of the theorem-proving program is the lack of editing strategies [1] to eliminate the large number of trivial deductions which cannot be excluded on purely logical grounds. Such deductions are trivial in the context of the problem at hand, but not in every context. It is therefore necessary to develop strategies of a "semantic" nature for specialized problem domains, to do this sort of editing. We have experimented with methods for using natural models for this purpose. However, it turns out that the use of anything but the most trivial structures involves a heavy cost in computation time, generally because the evaluation of satisfiability relative to a structure is a lengthy process (even for many-sorted models). It now appears that such editing applications can be achieved by working with operators on (partial) descriptions of models. (For the purpose of this discussion, a partial description of a model is a specification of some of those statements that are true of the model and some of the statements that are false). Preliminary experiments show that problems to do with the Advice Taker Project [6] or correctness of computer progams are fruitful areas in which to apply these strategies.

In addition to the development of the semantic editing strategies, other lines of research currently being pursued include the following.

(i) Addition of a procedure for question answering. It has recently been shown that the procedure [7] constructs interpolation formulas in the sense of [8].

(ii) Construction of a system for checking proofs of properties of computer programs, where the proofs are given in a form similar to that described in [9].

(iii) Experiments with the reduction of second order proof provability within two-sorted first-order theories[10].

REFERENCES

[1] Allen, J. and Luckham, D., "An Interactive Theorem-Proving Program", Machine Intelligence 5, (Meltzer, B. and Michie, D., Eds.), Edinburgh University Press, March 1970. Also, AIM-103.

[2] Kleburtz, R., Luckham, D., "Compatibility of Refinements of the Resolution Principle", submitted to JACM.

[3] Kleburtz, R., Luckham, D., "Compatible Strategies for the Resolution Principle", forthcoming A.I. Memo.

[4] Robinson, G., and Wos, L., "Paramodulation and Theorem- Proving in First-Order Theories with Equality", Machine Intelligence IV, D. Michie (ed), Edinburgh University Press, 1969.

[5] Hall, M., "The Theory of Groups", MacMillan, New York, 1959.

[6] McCarthy, J., "Programs with Common Sense", in M. Minsky (ed), Semantic Information Processing, MIT Press, Cambridge, 1968.

[7] Luckham, D., Nilsson, N., "Extracting Information from Resolution Proof Trees", submitted to the Artificial Intelligence Journal, and forthcoming A.I. Memo.

[8] Slagle, J., "Interpolation Theorems for Resolution in Lower Predicate Calculus", JACM, Vol. 17, No. 3, pp. 535-542.

[9] Burstall, R., "Formal Description of Program Structure and Semantics in First-Order Logic", Machine Intelligence 5, Edinburgh University Press, pp. 79-98.

[10] Skolem, T., "Reduction of Axiom Systems with Axiom Schemes to Systems with only Simple Axioms", Dialectica, Vol. 12, pp. 443-450, 1958.

1.5 Mathematical Theory of Computation (Robert Floyd, Donald Knuth, Zohar Manna, John McCarthy)

1.5.1 Recent Research

Substantial work has been done in developing techniques, using the methods of Floyd and Manna, for proving properties of algorithms. Considering the increased interest in the class of parallel algorithms, Ashcroft and Manna (1970) have extended these techniques for simple parallel programs. Although the programs considered are syntactically simple, they do exhibit interaction between asynchronous parallel processes. The formalization can be extended to more complicated programs. The method is based on a transformation of parallel progams into non-deterministic programs, the properties of which have been formalized in Manna (1970a). The non-deterministic programs are in general much larger than the parallel programs they correspond to. A simplification method is therefore presented which, for a given parallel program, allows the construction of a simple equivalent parallel program, whose corresponding non-deterministic program is of reasonable size. Further research is proceeding, emphasizing practical applications in such areas as time-sharing and multi-processing.

Manna (1970b) demonstrates conclusively that all properties regularly observed in programs (deterministic or non-deterministic) can be formulated in terms of a formalization of 'partial correctness'. Ashcroft (1970) 'explains' this by formulating the notion of an intuitively 'adequate' definition (in predicate calculus) of the semantics of a language or a program. He shows the relationship between a formalization or partial correctness of a program and an 'adequate' logical definition of its semantics. These two works give a general theory unifying the various logical approaches including those of Burstall, Cooper, Floyd, and Manna.

Manna and McCarthy (1970) formalize properties of Lisp-like programs using partial function logic, where the partial functions occurring in the formulas are exactly those computed by the programs. They distinguish between two types of computation rules -- sequential and parallel. McCarthy is trying to further develop axiomatic theories to handle 'undefinedness' in a natural way. Among other things, it may avoid paradoxical statements.

Igarashi (1970) has developed axiomatic methods for the semantics of Algol-like languages, mainly based on his earlier studies, but allowing the methods of Floyd to be carried out within the formalism. A metatheorem is included which can be interpreted as a proof of correctness of a conceptual compiler for the programs treated by the formalism.

Manna and Waldinger (1970) outlined a theorem-proving program approach to automatic program synthesis. In order to construct a program satisfying certain specifications, a theorem induced by those

specifications is proved, and the desired program is extracted from the proof. The use of the induction principle to construct programs with recursion is explored in some detail.

Other theoretical research in progress is mainly orientated towards practical applications. For example, Ashcroft, Manna and Pnueli have extended the class of schemas for which various properties are decidable. (These results give the decidability of the equivalence problem for Ianov schemas as a trivial case). Other work, by graduate students, is directed towards finding more powerful methods of proving equivalence of programs (Ness), detecting parallelism in sequential programs (Cadiou), and proving correctness of translators (Morris).

Currently our main emphasis is on preliminary studies for the construction of an interactive verification system. We wish to develop a practical system for proving programs correct that will be powerful enough to handle real programs.

1.5.2 Proposed Research

In the following we outline several research topics that we wish to undertake in the near future. Note that most of these topics are already being actively pursued.

1. To develop further the theory of equivalence, termination and correctness of computer programs.

2. To develop further the theory of semantic definition of programming languages, the formal description of translation algorithms, and the correctness of compilers.

3. To try to develop a theory of parallel processes adequate to prove their correctness and especially their mutual non-interference.

4. To develop a formal system of logic in which proofs of termination, equivalence, correctness, and non-interference can be conveniently express.

5. To pursue whatever new theoretical avenues appear likely to contribute to the goal of making checkout by proving correctness practical.

6. As soon as possible, Stanford graduate students in computer science will be asked to prove some of their programs correct as part of their course work so as to check out the techniques developed.

REFERENCES

[1] E. A. Ashcroft (1970), "Mathematical Logic Applied to the Semantics of Computer Programs", Ph.D. Thesis, to be submitted to Imperial College, London.

[2] E. A. Ashcroft and Z. Manna (1970), "Formalization of Properties of Parallel Programs", Stanford Artificial Inelligence Project, Memo AIM-110, to appear in Machine Intelligence 6, D. Michie (ed.), Edinburgh Univ. Press.

[3] S. Igarashi (1970), "Semantics of Algol-like Statements", Stanford Artificial Intelligence Project, Memo AIM-129.

[4] Z. Manna (1970a), "The Correctness of Non-determnistic Programs", Artificial Intelligence Journal, Vol. 1, No. 1.

[5] Z. Manna (1970b), "Second Order Mathematical Theory of Computation", Proc. ACM Symposium on Theory of Computing, May 1970.

[6] Z. Manna and J. McCarthy (1970), "Properties of Programs and Partial Function Logic", in Machine Intelligence 5, Edinburgh University Press.

[7] Z. Manna and R. Waldinger (1970), "Towards Automatic Program Synthesis", to appear in Comm. ACM.

1.6 Representation Theory (John McCarthy)

McCarthy will continue his investigations of ways of formally describing situations, laws giving the effects of actions, and laws of motion. New axiomatizations of knowledge and "can" are in the works.

Recent developments include work performed during a visit to Stanford by Erik Sandewall of Uppsala University on expressing natural language information in predicate calculus [1] and work by McCarthy on languages in which not all sentences have truth values.

Recent work in mathematical theory of computation by McCarthy, Ashcroft, and Manna on parallel and indeterminate computations and the correctness of non-halting programs has a direct application to representation theory because it permits proofs of correctness of strategies while processes other than the activity of the machine are going on. The proof checker development under the Mathematical Theory of Computation project will also promote this.

In the next period, work in representation theory will be carried out by McCarthy, Patrick Hayes, possibly David Luckham, and by graduate students.

REFERENCE

[1] E. Sandewall, "Representing Natural-Language Information in Predicate Calculus", AIM-128, Stanfor A. I. Project, July 1970.

1.7 COMPUTER SIMULATION OF BELIEF SYSTEMS [Kenneth Colby, Frank Hilf, Malcolm Newey, Roger Schank, Dave Smith, Larry Tesler, and Sylvia Weber]

Kenneth Mark Colby, M.D., who is a Senior Research Associate in the Computer Science Department, terminated his private practice of psychiatry to devote full time to investigations in this area of computer simulation. The National Institute of Mental Health sponsored two projects under Dr. Colby's direction. One of these is a Research Career Award and the other is a research project which continues the investigations in which his group has been engaged for the past seven years.

Introduction and Specific Aims

The clinical problems of psychopathology and psychotherapy require further investigation since so little is known about their essential processes. Some of this ignorance stems from a lack at a basic science level of dependable knowledge regarding higher mental processes such as cognition and affect. The research of the project attempts to approach both the clinical and basic science problems from the viewpoint of information-processing models and computer simulation techniques. This viewpoint is exemplified by current work in the fields of cognitive theory, attitude change, belief systems, computer simulation and artificial intelligence.

The rationale of our approach to these clinical problems lies in a conceptualization of them as information-processing problems involving higher mental functions. Computer concepts and techniques are appropriate to this level of conceptualization. Their success in other sciences would lead one to expect they might be of aid in the areas of psychopathology and psychotherapy.

The specific aims of this project relate to a long-term goal of developing more satisfactory explicit theories and models of psychopathological processes. The models can then be experimented with in ways which cannot be carried out on actual patients. Knowledge gained in this manner can then be applied to clinical situations.

Methods of Procedure

We have now gained considerable experience with methods for writing programs of two types. The first type of program represents a computer model of an individual person's belief system. We have constructed two versions of a model of an actual patient in psychotherapy and we are currently writing programs which simulate the belief systems of two normal individuals. We have also constructed a model of a pathological belief system in the form of an

artificial paranoia. A second type of program represents an interviewing program which attempts to conduct an on-line dialogue intended to collect data regarding an individual's interpersonal relations. We have written two such inteviewing programs and at present we are collaborating with psychiatrists in writing a program which can conduct a diagnostic psychiatric interview.

A computer model of a belief system consists of a large data-base and procedures for processing the information it contains. The data-base consists of concepts and beliefs organized in a structure which represents an individual's conceptualization of himself and other persons of importance to him in his life space. This data is collected from each individual informant by interviews. Verification of the model is also carried out in interviews in which the informant is asked to confirm or disconfirm the outcome of experiments on the particular model which represents his belief system. Because of the well-known effects of human interviewer bias, the process of data-collection and verifications should ideally be carried out by on-line man-machine dialogues and this is a major reason for our attempt to write interviewing programs. However, the difficulties in machine utilization of natural language remain great and until this problem is reduced we must use human interviewers.

We have written one type of therapeutic interactive program which is designed to aid language development in nonspeaking autistic children. We have used it for the past two years on eighteen children with considerable success (80% linguistic improvements). We intend to continue using this program and to instruct professionals in psychiatry and speech therapy in how to write, operate and improve such therapy programs for specific conditions.

Significance of this Research

This research has significance for the psychiatric, behavioral and computer sciences.

Psychiatry lacks satisfactory classifications and explanations of psychopathology. We feel these problems should be conceptualized in terms of pathological belief systems. Data collection in psychiatry is performed by humans whose interactive effects are believed to account for a large percentage of the unreliability in psychiatric diagnosis. Diagnostic interviewing should ideally be conducted by computer programs. Finally, the process and mechanisms of psyohotherapy are not well understood. Since experimentation on computer models is more feasible and controllable than experimentation on patients, this approach may contribute to our understanding of psychotherapy as an information-processing problem.

It is estimated that 90% of the data collected in the behavioral sciences is collected through interviews. Again, a great deal of the variance should be reduced by having consistent programs conduct

interviews. Also, this research has significance for cognitive theory, attitude change and social psychology.

Computer science is concerned with problems of man-machine dialogue in natural language, with optimal memory organization and with the search problem in large data-structures. This research bears on these problems as well as on a crucial problem in artificial intelligence, i.e., inductive inference by intelligent machines.

Collaboration

We are collaborating with two psychiatric centers for disturbed children and a local VA hospital. We are also collaborating with residents in the Department of Psychiatry and with graduate students in computer science, psychology, education and electrical engineering.

References

1. Colby, K.M., "Experimental Treatment of Neurotic Computer Programs", Archives of General Psychiatry, 10, 220-227 (1964).

2. Colby, K.M. and Gilbert, J.P., "Programming a Computer Model of Neurosis," Journal of Mathematical Psychology, 1, 405-417 (1964).

3. Colby, K.M. "Computer Simulation of Neurotic Processes", In Computers in Biomedical Research, Vol. I., Stacey, R.W. and Waxman, B., Eds., Academic Press, New York (1965).

4. Colby, K.M., Watt, J. and Gilbert, J.P. "A Computer Method of Psychotherapy," Journal of Nervous and Mental Disease, 142, 148-152 (1966).

5. Colby, K.M. and Enea, H. "Heuristic Methods for Computer Understanding in Context-restricted On-line Dialogues", Mathematical Biosciences, Vol. I, 1-25 (1967).

6. Colby, K.M. "Computer Simulation of Change in Personal Belief Systems," Behavioral Science, 12, 248-253 (1967).

7. Colby, K.M. "A Programmable Theory of Cognition and Affect in Individual Personal Belief Systems," In Theories of Cognitive Consistency, (Abelson, R., Aranson, E., McGuire, W., Newcomb, T., Tannebaum, P. Eds.,) Rand-McNally, New York, N.Y. (1968).

8. Colby, K.M. and Enea, H. "Inductive Inference by Intelligent Machines," Scientia, 103, 1-10 (1968).

9. Tesler, L., Enea, H, and Colby, K.M. "A Directed Graph Representation for Computer Simulation of Belief Systems," Mathematical Biosciences, 2, 19-40 (1968).

10. Colby, K.M. "Computer-aided Language Development in Nonspeaking Autistic Children." Technical Report, No. CS 85 (1967), Stanford Department of Computer Science. Archives of General Psychiatry, 19, 641-651 (1968).

11. Enea, H. "MLISP". Technical Report, CS 92 (1968), Stanford Department of Computer Science.

12. Colby, K.M., Tesler, L., Enea, H. "Search Experiments with the Data Base of Human Belief Structure", Proc. of the International Joint Conference on Artificial Intelligence, Washington, D.C., (Walker and Norton, Eds.), 1969.

13. Colby, K.M., Tesler, L. and Enea, H. "Experiments with a Search Algorithm on the Data Base of a Human Belief Structure," Stanford Artificial Intelligence Project Memo AIM-94, August 1969. Proceedings of the International Joint Conference on Artificial Intelligence, Walker and Norton (Eds.), 1969.

14. Colby, K.M. and Smith, D.C., "Dialogues Between Humans and An Artificial Belief System," Stanford Artificial Intelligence Project Memo AIM-97. Proceedings of the International Joint Conference on Artificial Intelligence, Walker and Norton (Eds.), 1969.

15. Colby, K.M. Critical Evaluation of "Some Empirical and Conceptual Bases for Coordinated Research in Psychiatry" by Strupp and Bergin. International Journal of Psychiatry, 7, 116-117 (1969).

16. Colby, K.M., Weber, S. and Hilf, F. "Artificial Paranoia," Stanford Artificial Intelligence Memo AIM-125, July 1970.

17. Colby, K.M., Hilf, F. and Hall, W. "A Mute Patient's Experience with Machine-Mediated Interviews", Stanford Artificial Intelligence Project Memo AIM-113, March 1970.

18. Hilf, F., Colby, K.M., Wittner, W. "Machine-Mediated Interviewing", Stanford Artificial Intelligence Project Memo AIM-112, March 1970.

19. Colby, K.M. "Mind and Brain, Again," Stanford Artificial Intelligence Project Memo AIM 116, March 1970.

20. Colby, K.M. and Smith, D.C. "Computer as Catalyst in the Treatment of Non-Speaking Autistic Children," Stanford Artificial Intelligence Project Memo AIM-120, April 1970.

21. Schank, R. " 'Semantics' in Conceptual Analysis," Stanford Artificial Intelligence Project Memo AIM-122, May 1970.

22. Smith, D.C. "MLISP," Stanford Artificial Intelligence Project Memo AIM-135, October 1970.

23. Schank, R.C. "Intention, Memory, and Computer Understanding," Stanford Artificial Intelligence Project Memo AIM-140, January 1971.

1.8 Facilities

By the time work on this proposal is to be initiated (1 July 1971) the computer facility will include the following.

Central Processors:	Digital Equipment Corporation PDP-10 and PDP-6
Primary Store:	65K Words of DEC Core (2us) 65K Words of Ampex Core (1us) 131K Words of new core (2us)
Swapping Store:	Librascope disk (5 million words, 22 million bits/second transfer rate)
File Store:	IBM 2314 disc file (leased)
Peripherals:	4 DEC tape drives, 2 mag tape drives, line printer, Calcomp plotter
Terminals:	15 Teletypes, 6 III displays, 1 ARDS display, 30 Data Disc displays, 3 IMLAC remote displays
Special Equipment:	Audio input and output systems, hand-eye equipment (2 TV cameras, 3 arms), remote-controlled cart

1.8.1 Processors

The system operates about 23.5 hours per day every day and is heavily loaded about 70% of this time. Lengthy compute-bound jobs such as computer vision, theorem-proving, and machine learning programs often bog down during high-load conditions to the point where it is difficult to complete segments of useful size in as much as an hour. With this problem in mind, we have explored alternative ways of increasing performance and concluded that the most productive approach will be to design and construct a special processor that is optimized for the class of problem we are dealing with.

Before the initiation of the proposal period, we expect to have completed the design of the new processor, submitted it to ARPA for review, received approval, and initiated fabrication. Funds already available under the existing contract should be sufficient to complete this project.

In the event that the design does not materialize in a timely way or that this project is disapproved, some other solution to the processing bottleneck will be needed. All known alternatives are more expensive and would exceed available funds. The budget of this proposal assumes that the new processor will be completed successfully.

We understand that one of the conditions for final approval of the new processor project will be that 50% of its capacity be made available to other participants in the ARPA net. This will be acceptable, however if the external usage makes substantial demands on other system elements (e.g. primary or secondary storage) it may be necessary to supplement these facilities. We have budgeted no funds for this.

1.8.2 Primary Store

We expect to have procured and installed core memory by the beginning of the proposal period, using funds made available under a supplement to this contract for Mathematical Theory of Computation. It may be desirable to augment this memory or replace some of the less reliable portions subsequent to the advent of the new processor.

1.8.3 Swapping Store

The Librascope disc file crashed some time ago, destroying half its capacity. Since the system is totally dependent on this unit for efficient operation, it is vulnerable to another crash. On behalf of the U.S. Government, Stanford is pressing a claim against Librascope (a division of Singer) regarding shortcomings in this equipment. It is hoped that there will be sufficient recovery under this suit to replace the disc with a more reliable swapping store.

1.8.4 File Store

Cost/performance considerations and the need to store somewhat more picture information for our computer vision research call for replacement of the 8-disc IBM 2314 file (leased) with a 4-disc version of the IBM 3330 (also leased).

1.8.5 Peripherals

Existing peripherals appear to be adequate with few exceptions. We have a need for a high speed printing device with general graphics capability (i.e. the capabilities of a plotter and the speed of a line printer). Existing devices in this class all seem to use special paper, which makes their operating costs too high. Devices that work with ordinary paper are under development by more than one manufacturer and we will likely want to procure one with available funds when they become available.

1.8.6 Terminals

Existing terminals appear adequate for the foreseeable future.

1.8.7 Special Equipment

Ongoing hand-eye research and other work in computer vision will require additional cameras with color vision capability, manipulators, and other instrumentation. Funds have been budgeted for these items.

BLANK PAGE

2. HEURISTIC PROGRAMMING PROJECT (Incorporating Heuristic DENDRAL)

(E.A. Feigenbaum, J. Lederberg, B. G. Buchanan, R. S. Engelmore)

TABLE OF CONTENTS

2.1 INTRODUCTION

Under previous ARPA contract support, the work of the Heuristic DENDRAL project has been focused on understanding the processes of scientific inference in problems involving the induction of hypotheses from empirical data, and on the implementation of a heuristic program for solving such problems in a real scientific setting. The Heuristic DENDRAL program now does a creditable, often extremely good, job of solving a variety of mass spectral analysis problems in organic chemistry.

It was necessary to invest the effort to construct this complex performance program as a foundation of understanding and a mechanism from which to build toward more interesting and important programs to do scientific theory formation. We have begun this building.

What we intend to do in the next two years is the subject of this proposal. The phase of ARPA support of the performance program writing and tuning (the Heuristic DENDRAL phase) will end with the expiration of the present contract period, though funds are being sought from NIH to continue that part of the work.

2.2 CHANGE OF PROJECT NAME

The focus of our work under ARPA support is expanding, and its nature is changing. Our desire to convey this explicitly leads us to want to change the project name to Heuristic Programming Project. This desire was reinforced by our observation that "Heuristic DENDRAL" has become, among computer scientists, a technical term referring to a

specific program, rather than a covering term for a group of people working on programs which model scientific thought processes.

2.3 PROPOSED WORK FOR NEW CONTRACT PERIOD: ITS RELATION TO LONG TERM GOALS

It is a paradox that the success of the Heuristic DENDRAL program in its chemical task area has tended to obscure our primary long-term motivations. Our scientific reports are by nature discrete entities, but the shape of the envelope function should not be lost sight of. Most of our plans for the coming period have arisen naturally from discussions in our research meetings over the last few years. Our problems, like most others, tend to have a natural time of ripeness; to pluck them earlier is to waste energy and resources and to incur frustration. It is toward an understanding of the overall shape of our work that we offer the following comments concerning goals.

Science and the work of scientists is our object of study. We seek our understanding not abstractly, as logicians and philosophers might, but concretely as scientists might (and sometimes do) about real science. We use the information processing viewpoint because this is the conceptualization that leads most directly from our observations about the processes of science and the behavior of scientists to computer programs that do scientific reasoning.

We seek to understand the formation of scientific theory as a means of organizing sets of observations; experiment as a technique for efficiently extracting new information about a universe of concern; how a scientific paradigm is used most effectively as framework for the conduct of routine scientific problem solving; how the framework may be altered when necessary; and what processes underly a scientific innovation.

Our specific plans for work on scientific theory formation are outlined in item 1, below.

We use the tools and concepts of artificial intelligence research, and as we write our programs and experiment with them, gaining some insights and understanding, we seek to pay back our debt by refining and adding to the fund of tools and concepts. As outlined in items 2, 3, and 4, below, our current research trajectory leads us to expect that we can contribute to an understanding of: the problem of representation of knowledge; the sources of generality and specialist expertise in problem solving programs; the nature of programming and the organization of heuristic programs.

1. Theory formation in Science

Five years ago we began our work by seeking a specific task environment in which to work, one of a complexity beyond the toy-like, yet simple enough so that we could formulate an approach with the conceptual tools we had at our command. We decided that the

problem area needed to have as its essence the inductive analysis of empirical data for the formation of explanatory hypotheses. This is the type of inference task that calls for the use of a scientific theory by a performance program, but not the formation of that theory. We did not have the insight, understanding, and daring at that time to tackle ab initio the problem of theory formation (and indeed it would have been foolhardy to do so then).

Now we feel the time is ripe for us to turn our attention in a major way to the problem of theory formation. Our understanding and our technical tools have matured along with the Heuristic DENDRAL program to the point where we now see clear ways to proceed. The effort, which began in a small way a few months ago is called Meta-DENDRAL.

As always, proper choice of task environment is crucial, but for us the choice was absolutely clear. The theory formation task most accessible to us is the task of forming mass spectral theory. Hence, the notion of building a level of programs "meta" to the DENDRAL performance program.

DENDRAL already contains an excellent mass spectral theory. We, therefore, have a clear idea of what a "correct answer" is like. DENDRAL's theory is represented in at least two different forms at present, so that we have a pretty good idea of the issues involved in representing mass spectral theory for a program. The Predictor program is an interesting kind of artificial experimental environment in which to perform certain kinds of internal "experiments" systematically, thereby generating a kind of systematic data that may not be available in the natural world. A theory language of notations, data structures, and primitive concepts (with which we are intimately familiar because we developed it) is available. People who are expert in the discovery of mass spectral theory are members of our research team. Many programs for manipulating mass spectral data have already been developed by us and are ready to be exploited as Meta DENDRAL tools.

The goal of the Meta-DENDRAL program is to infer the theory that the performance program (Heuristic DENDRAL) needs to solve problems of mass spectral analysis.

The following table attempts to sketch some differences between the programs at the performance level and the meta-level.

	Heuristic DENDRAL	Meta-DENDRAL
Input	The mass spectrum of a molecule whose structure is not known (except, of course, in our test cases)	A large number of recorded mass spectra and the associated (known) molecular structures.

Output	A molecular structure inferred from the data	A set of cleavage and rearrangement rules constituting a subset of the theory of mass spectrometry
Example	Uses alpha-carbon fragmentation theory rules in planning and in validation	Discovers (and validates) alpha-carbon fragmentation rules in a space of possible patterns of cleavage. Uses set of primitive concepts but does not invent new primitives.

In our view, the continuity evident in this table reflects a continuity in the processes of inductive explanation in Science. Moves toward meta-levels of scientific inference are moves toward encompassing broader data spaces and constructing more general rules for describing regularities in the data.

The number of possible ways of searching for regularities in a data space like that of Meta-DENDRAL's (and correspondingly, the number of generalizations that may be valid) is enormous. Consequently heuristic search strategies and path evaluations are necessary to constrain the search to subspaces that appear to be most fruitful by some (heuristic) criteria. We can not be too specific now about what shape these strategies and heuristics will eventually take, since this is the object of the research.

Two global organizations are being studied now. The first, in a rudimentary state, is very "human-like" and results from an attempt to get insights into the process by mimicking the inductive processes used by one of our chemist collaborators. This is an "incremental" type of strategy in which a rule (or part of a rule) is formulated on the basis of a very small number of examples, and is modified as additional cases are considered for which the rule fails partly or wholly.

The second is more "Machine-like" in the size of the data subset it is able to deal with at one time and in its systematic approach to rule finding. Each structure-spectrum pair is subjected to a bond-by-bond examination for evidence of cleavage. Bond-pair cleavages and then bond-triplet cleavages are searched for. Evidence for group migrations in conjunction with the various cleavages is sought (the generator treats hydrogen migrations first, more complex groups later). Validated events and the molecular context in which they occured are coded (described) in a language of primitive processes for the next stage of processing. This critical step is a search for regularity in the set of processes and contexts. It can be carried out in very many ways. A detailed understanding of the nature of such searches, and the effect of different search strategies, is an important part of our study. Regularities are expressed as tentative theory-rules in our situation-action rule representation (developed for Heuristic DENDRAL's Predictor).

Success of these tentative generalizations at predicting events for new data validates the rules. No running program yet exists which implements this scheme, though pieces of program are beginning to be written. The whole scheme is likely to undergo considerable evolution as we get more deeply into the effort.

2. Representation

We have seen in our previous work that the form in which knowledge about the (DENDRAL) world is represented is crucial to effective problem solving and to augmenting the knowledge structure for improved performance. Because of this, because the problem of representation is important to artificial intelligence research, and because we felt that the problem was more accessible to us than to most other groups (since we have a running problem solving program that uses and manipulates a complex body of knowledge), we have devoted considerable effort to representation.

We propose to continue these studies; indeed this is unavoidable in connection with the Meta-DENDRAL research. A substantial piece of work, begun in the current period, will be pushed. The work aims at the automatic re-representation of the knowledge of mass spectometry held by the Predictor in its "natural" process-oriented form to the more recent (and more satisfactory) production-rule form. The experiment involves trying to do this mainly by inductive techniques rather than by logical derivations. The Structure Generator will be used to produce appropriate and systematic input to the "old" Predictor to produce an artificial data space from which inductions to the "new" representation will be made. This effort obviously has considerable linkage to the Meta-DENDRAL work previously discussed.

3. Generality and Problem Solving

Transformation of representation is closely associated with the concept of specializing knowledge for its most efficient use in solving specific classes of problems. Our Planning Rule Generator already does this for a particular supra-family of mass spectral/chemical problems, the saturated acyclic monofunctional compounds. That is, "specialists" for particular chemical families are automatically inferred from alpha-carbon fragmentation theory.

A.I. Memo 131 (23) discusses expertise and problem-solving specialists in contrast to general problem solving mechanisms at the performance level that select appropriate specialists. This is a path in the search for generality of problem solvers that has not previously been explored in detail (though Moses' SIN program lies on the path). We plan to do a more detailed exploration, for which A.I. Memo 131 was our opening statement.

4. The Nature of Programs and the Organization of Heuristic Programs

One of the most important sources of transfer from our present work to our future work and to the work of others is likely to result from a detailed examination of the DENDRAL programs as an organized sequence of manipulations on a symbolic world internal to the LISP "machine". In our research discussions, we return repeatedly to problems involved in trying to understand systematically that universe of entities known as computer programs, and in particular the subclass of heuristic programs. Why?

First, as builders of perhaps the largest heuristic program that exists, we are forever frustrated that our next steps are not more readily accomplished than our last steps, that there remains undiscovered something systematic and understandable that will permit next steps to be made more scientifically and less artisan-like than previous steps.

Second, the programming task itself presents a problem domain worthy of intense application-oriented research by the A.I. community. It is almost certainly true that two or three decades hence most computer programs (as we know them today) will be written by computer program. The necessary initial explorations should begin now (as indeed they have begun at a few places).

Third, our work is primarily concerned with heuristic programs and these present particular problems and challenges. At some level in the organiztion of our programs we are writing heuristic procedures and not merely routine symbol manipulations (for example, search cut-off decisions as opposed to some list-structure reorganization). But there is so little heuristic programming science to draw upon in engineering specific heuristic procedures! To the extent that heuristic programming is a science at all, it is an experimental science (invent, organize, implement, try, observe, interpret; reorganize, invent more, try again, observe and interpret; and so on). The processes of the automatic heuristic program writer (or at least programming assistant) may be similar to the processes of the empirical scientist, the problem domain of our primary interest.

For all of these reasons, we plan to invest some of our time and resources in a exploratory effort to better understand programs, program construction, and in particular the organization of heuristic programs. Whether we pursue our quest for understanding by program writing is not now clear. It will probably depend on what individuals, particularly graduate students, become interested in pursuing these questions.

5. New Artificial Intelligence Application

During the current period, we have spent much effort in search of a new application in which the techniques and concepts of artificial intelligence research can be applied to a problem of importance to

Science. To serve adequately, the problem domain must be such that complex reasoning processes play a significant role in the discovery of problem solutions (interestingly, many scientific tasks do not have this property). In line with our general inclinations, problem domains of primary interest to us are those involving inductive generalization and hypothesis formation from sets of empirical data. There are many other characteristics that a problem area must have if it is to be of interest and use to us, but we will not discuss these here.

During the remainder of the current period, a new application area will probably be selected. We intend that the project support this work during the period of problem formulation and the period of initial testing for feasibility. If the idea is viable, sustaining support to bring the application to fruition will probably be sought from NSF or NIH.

2.4 HISTORICAL SYNOPSIS

Work began in 1965 and has been supported by ARPA contract funds for artificial intelligence research at Stanford since that time. A specific task environment was decided upon as a context for studying scientific hypothesis formation. This was: the induction of hypotheses about organic molecular structure from physycal spectra of molecules (initially mass spectra, later including nuclear magnetic resonance spectra as auxiliary data). The heuristic program written to solve such problems consists broadly of two phases: hypothesis generation and hypothesis validation.

The hypothesis generation phase is a heuristic search in which a combinatorial space of possible candidate molecular structures is severely constrained by: (a) heuristics that define plausibility of structures in terms of their natural stability and (b) a search plan inferred from the problem data using task-specific pattern analysis heuristics that are based on mass spectral theory. The chemical knowledge (theories, conjectures, facts, heuristics) used by the program has been elicited from professional mass spectrometrists by a reasonably systematic technique, often employing interaction between the human chemist and the program. In some instances, the heuristics used for inferring the search plan can be deduced by one of our programs, without recourse to our chemist collaborators.

In the hypothesis validation phase, candidate hypotheses are evaluated by a two step process: (a) a program using a complex theory of mass spectra generates a prediction against problem data, makes a final discarding of some candidates, and ranks those remaining.

This program is written in LISP. In terms of bytes of core memory for the IBM 360 (including necessary binary program space and adequate free storage space) a typical "working" core image will occupy over one million bytes (usually run in three 350K job steps).

Other DENDRAL code used in the past and sometimes in the present constitutes another half million bytes.

The program has solved hundreds of structure determination problems. For the supra-family of organic molecules upon which we have focused most intensely (saturated, acyclic, monofunctional compounds) the performance is extremely good, measured in time-to-solution and in quality of solutions, even compared with the best human performance. In other families and supra-families, the performance is sometimes good, sometimes novice-like. The basic processes are completely general, however, so that increments of new chemical knowledge will readily give rise to better performance on a broader range of problems.

This work has been reported to the computer science community in the following publications: [7], [10], [16], [17], [23]. Since the program is of considerable interest as an application in chemistry, we have produced a series of papers for the chemical literature entitled "Applications of Artificial Intelligence for Chemical Inference", of which six papers have appeared [14], [15], [19], [20], [21], [22], and two more are in progress. The work has been discussed in terms of philosophy of science in [18].

2.5 VIEWS OF OTHERS CONCERNING THIS RESEARCH

The publication of this work has engendered considerable discussion and comment among computer scientists and chemists. Professor H. Gelernter (SUNY, Stony Brook), at an SJCC 1970 panel of the use of computers in science gave an extended discussion of the program, in which he said that it was the first important scientific application of artificial intelligence. Dr. W. H. Ware (RAND Corporation) in a recent article entitled "The Computer in Your Future" in the collection Science and Technology in the World of the Future (A. B. Bronwell, ed., Wiley Press, 1970) said:

> "Thus, much of engineering will be routinized at a high level of sophistication, but what about science? An indication of what is coming at a higher level of intellectual performance is a computer program called Heuristic DENDRAL, which does a task that a physical chemist or biologist concerned with organic chemistry does repeatedly."

Professor J. Weizenbaum of MIT, in an undergraduate computer science curriculum proposal for MIT entitled "A First Draft of a Proposal for a New Introductory Subject in Computer Science (September 1970)" included Heuristic DENDRAL in his "group 4" series (three lectures) entitled Great Programs; and he said recently (personal communication), commenting on recent work and plans,

"I see the work you are now beginning as a step in the direction of composing explicit models of just such programs (that build expertise in an area). The implications of a success in such an effort are staggering. I therefore believe your effort to be worthy of very considerable investment of human and financial resources."

In his paper presented at the Sixth International Machine Intelligence Workshop, Professor Saul Amarel (Rutgers University) used Heuristic DENDRAL to illustrate a point about programs which use theoretical knowledge. He wrote:

"The DENDRAL system provides an excellent vehicle for the study of uses of relevant theoretical knowledge in the context for formation problems." [from "Representations and Modeling in Problems of Program Formation", Saul Amarel. In Machine Intelligence 6 (B. Meltzer and D. Michie, eds.) Edinburgh University Press (in press)].

Dr. T. G. Evans (Air Force Cambridge Research Labs), President of the ACM SIGART, in introducing a talk on Heuristic DENDRAL at the 1970 FJCC meeting of SIGART, called the program "probably the smartest program in the world" (and followed this with the interesting observation that this program had probably received a more sustained and intense effort than any other single program in the history of the artificial intelligence field). At a practical level, a firm of British computer science consultants headed by Professor D. Michie of the University of Edinburgh requested and obtained permission to adapt and market the use of the Heuristic DENDRAL program to a British chemical company. A mass spectometry laboratory at the University of Goteberg in Sweden, headed by an eminent mass spectrometrist, has asked for listings and help in beginning the use of the program there. On a recent visit to Stanford representatives from one of the leading mass spectrometer manufacturers (Varian-MAT, Germany) also requested listings of the program.

2.6 REVIEW OF WORK OF THE CURRENT PERIOD

In the previous proposal, we outlined work to be undertaken during the current period. There has been substantial progress on much of this work, though we are only a year into the current period. Some of the work has not been attempted because its time did not seem to be ripe.

We have already mentioned that dramatic progress was made in the improvement of the performance of the program as an application to chemistry. Our first paper on ringed structures was published and more complex work (on steroids) is now being done. Other functional groups were added to the Planner. N.M.R. analysis was brought to bear in a meaningful way at the level of planning. As we indicated earlier, the program is moving to the stage at which it can be exported to the scientific community.

Our work on representation of knowledge, so dominant a theme in our previous proposal, has been multi-faceted. The knowledge we are dealing with is the theory of the mass spectrometric processes of fragmentation and recombination. This theory, the basis of our Predictor, has been rerepresented in a particular production rule ("situation-action" rule) form as a preliminary to writing programs that will perform this representational transformation automatically.

This new representation is also the internal representation for the knowledge acquired by the interactive Dialog program for eliciting knowledge about chemical structures and processes from practitioners. A rather specialized chemistry language in which to conduct this dialog has been created, as well as the interpreter for that language. This work will be the subject of an A.I. Memo later in the contract period.

There are at least two aspects to the problem of the effective use of general knowledge about a world by a problem solver: at what point in the problem solving process the knowledge is deployed, and the representation chosen for its deployment at that point. We have done substantial work on both of these problems With respect to these problems, the most dramatic event of this period was the construction of the Planning Rule Generator (described in A.I. Memo 131 [23]). This is the program that deduces, from some general first-order mass spectral theory that is basic to the Predictor's activity, the specific family-related pattern recognition heuristics used in the Planning process. It deduces "experts" for specific chemical families for deployment early in the hypothesis generation phase. Of the other experiments done with respest to point of deployment of knowledge, some have had spectacular effect in search reduction, as for example, the introduction of the N.M.R. analysis during Planning rather than as a terminal evaluaion step. (The former is discussed in A.I. Memo 131 (23), the latter in [19].)

Scientific reports on our experiments with representation and design will be forthcoming as A.I. Memos during the Spring and Summer of 1971.

In this period, also, we have been able to formulate and begin the groundwork for the next period's work on meta-DENDRAL, discussed earlier.

2.7 HEURISTIC DENDRAL AS APPLICATION TO CHEMISTRY: POSSIBLE WIH SUPPORT

It should be clear from the earlier sections of this proposal that we have demonstrated the feasibility (at least) of applying techniques of artificial intelligence research to structure determination problems in organic chemistry in a meaningful and practical way. Feasibility, however, is not realization. A very considerable amount of hard work by chemists and DENDRAL programmers remains to be done to make a comprehensive practical scientific tool. ARPA is not being

asked to support this part of the research and development.

An NIH grant application was submitted last April and given an extraordinarily comprehensive scientific review by NIH. The application was approved and is now awaiting funding by the Division of Research Resources, which funds computer facilities, mass spectrometer facilities and other expensive instrumental tools of bio-medical research, along with research into methods for their most effective use.

The NIH proposal calls for:

1. Experimental work with a new mass spectrometer.
2. Organizing and programming existing and newly-developed knowledge about mass spectrometry to improve the breadth and quality of the performance of the Heuristic DENDRAL program.
3. The control of a mass spectrometer with a gas-liquid chromatograph in real-time by the Heuristic DENDRAL program, such that the whole system is solving a problem rather than merely collecting data for later analysis.
4. Meta-DENDRAL research on theory formation in mass spectrometry (a very small fraction since this work is advanced A.I. research and is central to the ARPA -sponsored effort).

2.8 COMPUTER FACILITIES

Fortunately, the project is blessed with excellent computer facilities at the moment, so that the only budgetery proposal that needs to be made in this regard is for the purchase of services and not for the development of a resource. Our programming is done almost entirely in Stanford 360/LISP.

On the 360/67 at the Computation Center's Facility, the following is available:

Remote Job Entry to Batch Partitions via the WYLBUR Text Editor, with job output available at the terminal.
Partition Sizes for Batch Partitions:
131K bytes in separate high-speed partition for diagnostic runs
280K bytes, normal partition size
411K bytes, large partition size
800K bytes, "giant" partition size, available on overnight runs
Interactive time-shared LISP interpreter and compiler, available under ORVYL time-sharing submonitor

On the 360/50 at the Medical School's ACME Computer Facility, the following is available under the ACME time-sharing monitor (non-swapping):

360/LISP (interpreter and compiler)

Amount of memory: up to a few hundred thousand bytes in the daytime operation, variable depending upon our immediate request, up to 1.8 million bytes of slow-speed core memory available at night and at off hours.

2.9 BUDGETARY NOTE CONCERNING COMPUTER TIME

The need to hold the overall annual budget constant as we move to the next contract period, coupled with the need to absorb expenditure increases in a variety of budgetary categories, necessitated the budgeting of reduced computer time expenditures from the budgeted $6000/ month of the present contract to $4000/ month. The possible adverse impact of this reduction can (hopefully) be mitigated by:

a. the use of some of the NIH grant funds (if our proposal is funded) for certain parts of the work,
b. use of the Artificial Intelligence Project's facilities for part of the work,
c. use of ARPA network facilities, where feasible and appropriate.

Thus, the fallback positions appear at present to be adequate.

2.10 BUDGETARY NOTE CONCERNING PERSONNEL

In addition to the people mentioned in the ARPA-supported budget (Feigenbaum, Lederberg, Buchanan, Engelmore, and graduate students), other project scientists are provided by the Stanford Mass Spectrometry Laboratory, and the Instrumentation Research Laboratory of the Genetics Department. Additional positions are requested in the NIH grant application to carry out tasks called for there.

BIBLIOGRAPHY

[1] J. Lederberg, "DENDRAL-64-A System for Computer Construction, Enumeration and Notation of Organic Molecules as Tree Structures and Cyclic Graphs", (technical reports to NASA, also available from the author and summarized in [12]).

- [1a] Part I. Notational algorithm for tree structures. (1965) CR.57029
- [1b] Part II. Topology of cyclic graphs (1965) CR.68898
- [1c] Part III. Complete chemical graphs: embedding rings in trees (1969)

[2] J. Lederberg, "Computation of Molecular Formulas for Mass Spectrometry", Holden-Day, Inc. (1964).

[3] J. Lederberg, "Topological Mapping of Organic Molecules", Proc. Nat. Acad. Sci., 53:1, January 1965, pp. 134-139.

[4] J. Lederberg, "Systematics of organic molecules, graph topology and Hamilton Circuits. A general outline of the DENDRAL system," NASA CR-48899 (1965).

[5] J. Lederberg, "Hamilton Circuits of Convex Trivalent Polyhedra (up to 18 vertices), Am. Math. Monthly, May 1967.

[6] G.L. Sutherland, "DENDRAL - A Computer Program for Generating and Filtering Chemical Structures", Stanford Artificial Intelligence Project Memo No. 49, February 1967.

[7] J. Lederberg and E.A. Feigenbaum, "Mechanization of Inductive Inference in Organic Chemistry", in B. Kleinmuntz (ed) Formal Representations for Human Judgment, (Wiley, 1968) (also Stanford Artificial Intelligence Project Memo No. 54, August 1967).

[8] J. Lederberg, "Online computation of molecular formulas from mass number." NASA CR-94977 (1968).

[9] E.A. Feigenbaum and B.C. Buchanan, "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry", in Proceedings, Hawaii International Conference on System Sciences, B.K. Kinariwala and F.F. Kuo (eds), January 1968.

[10] B.G. Buchanan, G.L. Sutherland, and E.A. Feigenbaum, "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry", in Machine Intelligence 4 (B. Meltzer and D. Michie, eds) Edinburgh University Press (1969), (also Stanford Artificial Intelligence Project Memo No. 62, July 1968).

[11] E.A. Feigenbaum, "Artificial Intelligence: Themes in the Second Decade", In Final Supplement to Proceedings of the IFIP 68 International Congress, Edinburgh, August 1968 (also Stanford Artificial Intelligence Project Memo No. 67, August 1968).

[12] J. Lederberg, "Topology of Molecules", in The Mathematical Sciences - A Collection of Essays, (ed.) Committee on Support of Research in the Mathematical Sciences (COSRIMS), National Academy of Sciences - National Research Council, M.I.T. Press, (1969), pp. 37-51.

[13] G. Sutherland, "Heuristic DENDRAL: A Family of LISP Programs", to appear in D. Bobrow (ed), LISP Applications (also Stanford Artificial Intelligence Project Memo No. 80, March 1969).

[14] J. Lederberg, G.L. Sutherland, B.G. Buchanan, E.A. Feigenbaum. A.V. Robertson, A.M. Duffield, and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference I. The Number of Possible Organic Compounds: Acyclic Structures Containing C, H, O and N". Journal of the American Chemical Society, 91:11 (May 21, 1969).

[15] A.M. Duffield, A.V. Robertson, C. Djerassi, B.G. Buchanan, G.L. Sutherland, E.A. Feigenbaum, and J. Lederberg, "Application of Artificial Intelligence for Chemical Inference II, Interpretation of Low Resolution Mass Spectra of Ketones", Journal of the American

Chemical Society, 91:11 (May 21, 1969).

RECENT PUBLICATIONS

[16] B.G. Buchanan, G.L. Sutherland, E.A. Feigenbaum, "Toward an Understanding of Information Processes of Scientific Inference in the Context of Organic Chemistry", in Machine Intelligence 5, (B. Meltzer and D. Michie, eds) Edinburgh University Press (1969). (also Stanford Artificial Intelligence Project Memo No. 99, September 1969).

[17] J. Lederberg, G.L. Sutherland, B.G. Buchanan, and E.A. Feigenbaum, "A Heuristic Program for Solving a Scientific Inference Problem: Summary of Motivation and Implementation", in Theoretical Approaches to Non-Numerical Problem Solving (R. Banerji and M.D. Mesarovic, eds.) Springer-Verlag, New York (1970). (Also Stanford Artificial Intelligence Project Memo No. 104, November 1969).

[18] C.W. Churchman and B.G. Buchanan, "On the Design of Inductive Systems: Some Philosophical Problems". British Journal for the Philosophy of Science, 20 (1969), pp. 311-323.

[19] G. Schroll, A.M. Duffield, C. Djerassi, B.G. Buchanan, G.L. Sutherland, E.A. Feigenbaum, and J. Lederberg, "Application of Artificial Intelligence for Chemical Inference III. Aliphatic Ethers Diagnosed by Their Low Resolution Mass Spectra and NMR Data". Journal of the American Chemical Society, 91:26 (December 17, 1969).

[20] A. Buchs, A.M. Duffield, G. Schroll, C. Djerassi, A.B. Delfino, B.G. Buchanan, G.L. Sutherland, E.A. Feigenbaum, and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference IV. Saturated Amines Diagnosed by their Low Resolution Mass Spectra and Nuclear Magnetic Resonance Spectra", Journal of the American Chemical Society, 92:23 (November 18, 1970).

[21] Y.M. Sheikh, A. Buchs, A.B. Delfino, B.G. Buchanan, G.L. Sutherland, E.A. Feigenbaum, and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference V. An Approach to the Computer Generation of Cyclic Structures. Differentiation Between All the Possible Isomeric Ketones of Composition C6H10O", Organic Mass Spectrometry (in press).

[22] A. Buchs, A.B. Delfino, A.M. Duffield, C. Djerassi, B.G. Buchanan, E.A. Feigenbaum, and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference VI. Approach to a General Method of Interpreting Low Resolution Mass Spectra with a Computer", Helvetica Chemica Acta, 53:6 (1970).

[23] E.A. Feigenbaum, B.G. Buchanan, and J. Lederberg, "On Generality and Problem Solving: A Case Study Using the DENDRAL Program", in Machine Intelligence 6 (B. Meltzer and D. Michie, eds) Edinburgh University Press (in press). (Also, Stanford Artificial

Intelligence Project Memo No. 131).

[24] B.G. Buchanan and T.E. Headrick, "Some Speculation about Artificial Intelligence and Legal Reasoning". Stanford Law Review, November, 1970. (Also, Stanford Artificial Intelligence Project Memo No. 123).

[25] B.G. Buchanan, A.., Duffield and A.V. Robertson, "An Application of Artificial Intelligence to the Interpretation of Mass Spectra". In Mass Spectrometry, 1970 (G.W. Milne, ed.) Wiley (in press).

BLANK PAGE

3.BUDGET

Budgets for the Artificial Intelligence (A.I.) and Heuristic Programming (H.P.) projects are given below for the next two fiscal years. It may be noted that the amounts allocated to salaries are the same for both years, even though inflation may be expected to take its toll. The budget will be maintained by permitting attrition to reduce staff size.

3.1 SUMMARY OF BUDGETS FOR CONTINUATION OF SD-183 (Fiscal Year 1972)

BUDGET ITEM	1 JUL 71 TO 30 JUN 72		TOTAL
	A.I.	H.P.	
Salaries	$467,526	$ 79,628	$ 547,154
Staff Benefits	70,051	11,931	81,982
Travel	30,500	3,700	34,200
Capital Equipment	108,000	- - -	108,000
Equipment Rental	50,319	5,400	55,719
Computer Time	- - -	46,081	46,081
Equipment Maintenance	40,000	- - -	40,000
Communications	14,400	1,500	15,900
Publications	14,000	1,600	15,600
Other Operating Expenses	32,942	1,800	34,742
Indirect Costs	322,262	48,360	370,622
Totals	$ 1,150,000	$ 200,000	$ 1,350,000

3.2 SUMMARY OF BUDGETS FOR CONTINUATION OF SD-183 (Fiscal Year 1973)

BUDGET ITEM	1 JUL 72 TO 30 JUN 73 A.I.	H.P.	TOTAL
Salaries	$467,526	$ 79,628	$ 547,154
Staff Benefits	76,908	13,084	89,992
Travel	30,500	3,700	34,200
Capital Equipment	79,855	- - -	79,855
Equipment Rental	56,700	5,400	62,100
Computer Time	- - -	44,200	44,200
Equipment Maintenance	40,000	- - -	40,000
Communications	14,400	1,500	15,900
Publications	14,000	1,600	15,600
Other Operating Expenses	32,942	1,800	34,742
Indirect Costs	337,169	49,088	386,257
Totals	$ 1,150,000	$ 200,000	$ 1,350,000

3.3 ARTIFICIAL INTELLIGENCE BUDGET

			1-JUL-71 TO 30-JUN-72	1-JUL-72 TO 30-JUN-73
I. TOTAL ARTIFICIAL INTELLIGENCE SALARIES			467,526	467,526
II. STAFF BENEFITS-				
13.9% to 8-31-71	10,831			
15.2% to 8-31-72	59,220	11,344		
16.7% to 8-31-73		65,064		
TOTAL STAFF BENEFITS			70,051	76,908
III. TRAVEL				
6 Foreign trips, 1200/ea.	7,200			
20 Trips east, 450/ea.	9,000			
5 Professional staff moves to Stanford, 1900/ea.	9,500			
Local travel	4,800			
TOTAL TRAVEL			30,500	30,500
IV. CAPITAL EQUIPMENT				
5-IBM 3336 Disk Packs	5,000			
Test Equipment (Arm and camera instrumentation)	70,000			
Color Equipment (Camera, mount, filters)	33,000			
Computer peripherals and Test Equipment		79,855		
TOTAL CAPITAL EQUIPMENT			108,000	79,855
V. EQUIPMENT RENTAL				
IBM Disk File and Packs (2314, 3330)	50,319			
IBM 3330 Disk File		56,700		
TOTAL EQUIPMENT RENTAL			50,319	56,700
VI. EQUIPMENT MAINTENANCE (based on past experience)			40,000	40,000

VII.	COMMUNICATIONS --------------------------- (Telephones, dataphones, teletypes)		14,400	14,400
VIII.	PUBLICATIONS COST (Past Experience)------		14,000	14,000
IX.	OTHER OPERATING EXPENSES---------------- (e.g. Office Supplies, Postage, Freight, Consulting, Utilities)		32,942	32,942
X.	INDIRECT COSTS			
	59% of salaries to 9-1-71	45,974		
	46% of modified direct costs thereafter (direct costs less capital equipment)	276,288		
	TOTAL INDIRECT COSTS-----------		322,262	337,169
	TOTAL ARTIFICIAL INTELLIGENCE BUDGET----- $		1,150,000	1,150,000

3.4 HEURISTIC PROGRAMMING BUDGET

			1-JUL-71 TO 30-JUN-72	1-JUL-72 TO 30-JUN-73
XI.	TOTAL HEURISTIC PROGRAMMING SALARIES ----		$ 79,628	79,628
XII.	STAFF BENEFITS-			
	13.9% to 8-31-71	1,845		
	15.2% to 8-31-72	10,086 2,002		
	6.7% to 8-31-73	11,082		
	TOTAL STAFF BENEFITS-------------		11,931	13,084
XIII.	TRAVEL			
	2 Foreign trips, $1200. ea.	2,400		
	2 Trips East, $450. ea.	900		
	Local Travel	400		
	TOTAL TRAVEL--------------------		3,700	3,700
XIV.	EQUIPMENT RENTAL (Wylbur Terminals-360/67)-		5,400	5,400
XV.	COMPUTER TIME (IBM 360/67 time)-----------		46,081	44,200
XVI.	COMMUNICATIONS(Telephones, Dataphones)-----		1,500	1,500
XVII.	PUBLICATIONS-------------------------------		1,600	1,600
XVIII.	OTHER OPERATING EXPENSES------------------		1,800	1,800
XIX.	INDIRECT COSTS			
	59% of salaries to 8-31-71	7,830		
	46% of modified direct costs thereafter (direct costs less computer time)	40,530		
	TOTAL INDIRECT COSTS-------------		48,360	49,088
	TOTAL HEURISTIC PROGRAMMING BUDGET------------		$200,000	200,000
XX.	TOTAL SD-183 BUDGET---------------------		$1,350,000	1,350,000

4. Cognizant Personnel

For contractual matters:

Office of the Research Administrator
Stanford University
Stanford, California 94305

Telephone: (415) 321-2300, ext. 2883

For technical and scientific matters regarding the Artificial Intelligence Project:

Prof. John McCarthy
Dr. Arthur Samuel
Mr. Lester Earnest
Computer Science Department
Telephone: (415) 321-2300, ext. 4971

For technical and scientific matters regarding the Heuristic programming project:

Prof. Edward Feigenbaum
Computer Science Department
Telephone: (415) 321-2300, ext. 4878

Prof. Joshua Lederberg
Genetics Department
Telephone: (415) 321-2300, ext. 5052

For administrative matters, including questions relating to the budget or property acquisition:

Mr. Lester D. Earnest
Mr. Norman Briggs
Computer Science Department
Stanford University
Stanford, California 94305

Telephone: (415) 321-2300, ext. 4971

APPENDIX A

PUBLICATIONS

Articles and books by members of the Stanford Artificial Intelligence Project are listed here by year. Only publications following the individual's affiliation with the Project are given.

1963

1. J. McCarthy, "A Basis for a Mathematical Theory of Computation," in P. Blaffort and D. Hershberg (eds.), Computer Programming and Formal Systems, North-Holland, Amsterdam, 1963.

2. J. McCarthy, "Towards a Mathematical Theory of Computation," in Proc. IFIP Congress 62, North-Holland, Amsterdam, 1963.

3. J. McCarthy (with S. Bollen, E. Fredkin, and J.C.R. Licklider), "A Time-Sharing Debugging System for a Small Computer," in Proc. AFIPS Conf. (SJCC), Vol. 23, 1963.

4. J. McCarthy (with F. Corbato and M. Daggett), "The Linking Segment Subprogram Language and Linking Loader Programming Languages," Comm. ACM, July, 1963.

1965

1. J. McCarthy, "Problems in the Theory of Computation," in Proc. IFIP Congress 65, Spartan, Washington, D.C., 1965.

1966

1. A. Hearn, "Computation of Algebraic Properties of Elementary Particle Reactions Using a Digital Computer," Comm. ACM, 9, pp. 573-577, August, 1966.

2. J. McCarthy, "A Formal Description of a Subset of Algol," in T. Steele (ed.), Formal Language Description Languages, North-Holland, Amsterdam, 1966.

3. J. McCarthy, "Information," Scientific American, September, 1966.

4. J. McCarthy, "Time-Sharing Computer Systems," in W. Orr (ed.), Conversational Computers, Wiley, 1966.

5. D. Reddy, "Segmentation of Speech Sounds," J. Acoust. Soc. Amer., August 1966.

1967

1. S. Brodsky and J. Sullivan, "W-Boson Contribution to the Anomalous Magnetic Moment of the Muon," Phys Rev 156, 1644, 1967.

2. J. Campbell, "Algebraic Computation of Radiative Corrections for Electron-Proton Scattering," Nuclear Physics, Vol. B1, pp. 238-300, 1967.

3. E. Feigenbaum, "Information Processing and Memory," in Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 4, U.C. Press, Berkeley, 1967.

4. J. Goodman, "Digital Image Formation from Electronically Detected Holograms," in Proc. SPIE Seminar on Digital Imaging Techniques, Soc. Photo-Optical Instrumentation Engineering, Redondo Beach, California, 1967.

5. J. Goodman, "Digital Image Formation from Electronically Detected Holograms," Applied Physics Letters, August 1967.

6. A. Hearn, "REDUCE, A User-Oriented Interactive System for Algebraic Simplification, Proc. ACM Symposium on Interactive Systems for Experimental Applied Mathematics, August 1967.

7. J. Lederberg, "Hamilton Circuits of Convex Trivalent Polyhedra," American Mathematical Monthly 74, 522, May 1967.

8. J. McCarthy, D. Brian, G. Feldman, and J. Allen, "THOR--A Display Based Time-Sharing System," AFIPS Conf. Proc., Vol. 30, (FJCC), Thompson, Washington, D.C., 1967.

9. J. McCarthy, "Computer Control of a Hand and Eye," in Proc. Third All-Union Conference on Automatic Control (Technical Cybernetics), Nauka, Moscow, 1967 (Russian).

10. J. McCarthy and J. Painter, "Correctness of a Compiler for Arithmetic Expressions," Amer. Math. Soc., Proc. Symposia in Applied Math., Math. Aspects of Computer Science, New York, 1967.

11. D. Reddy, "Phoneme Grouping for Speech Recognition," J. Acoust. Soc. Amer., May, 1967.

12. D. Reddy, "Pitch Period Determination of Speech Sounds," Comm. ACM, June, 1967.

13. D. Reddy, Computer Recognition of Connected Speech," J. Acoust. Soc. Amer., August, 1967.

14. A. Samuel, "Studies in Machine Learning Using the Game of Checkers, II-Recent Progress," IBM Journal, November, 1967.

15. G. Sutherland (with G.W. Evans and G.F. Wallace), Simulation Using Digital Computers, Prentice-Hall, Engelwood Cliffs, N.J., 1967.

1968

1. E. Feigenbaum, J. Lederberg and B. Buchanan, "Heuristic Dendral", Proc. International Conference on System Sciences, University of Hawaii and IEEE, University of Hawaii Press, 1968.

2. E. Feigenbaum, "Artificial Intelligence: Themes in the Second Decade", Proc. IFIP Congress, 1968.

3. J. Feldman (with D. Gries), "Translator Writing Systems", Comm. ACM, February 1968.

4. J. Feldman (with P. Rovner), "The Leap Language Data Structure", Proc. IFIP Congress, 1968.

5. R. Gruen and W. Weiher, "Rapid Program Generation", Proc. DECUS Symposium, Fall 1968.

6. A. Hearn, "The Problem of Substitution", Proc. IBM Summer Institute on Symbolic Mathematics by Computer, July 1968.

7. D. Kaplan, "Some Completeness Results in the Mathematical Theory of Computation", ACM Journal, January 1968.

8. J. Lederberg and E. Feigenbaum, "Mechanization of Inductive Inference in Organic Chemistry", in B. Kleinmuntz (ed.), Formal Representation of Human Judgment, John Wiley, New York, 1968.

9. J. McCarthy, "Programs with Common Sense" in M. Minsky (ed.), Semantic Information Processing, MIT Press, Cambridge, 1968.

10. J. McCarthy, L. Earnest, D. Reddy, and P. Vicens, "A Computer with Hands, Eyes, and Ears", Proc. AFIPS Conf. (FJCC), 1968.

11. K. Pingle, J. Singer, and W. Wichman, "Computer Control of a Mechanical Arm through Visual Input", Proc. IFIP Congress 1968, 1968.

12. D. Reddy, and Ann Robinson, "Phoneme-to-Grapheme Translation of English", IEEE Trans. Audio and Electroacoustics, June 1968.

13. D. Reddy, "Computer Transcription of Phonemic Symbols", J. Acoust. Soc. Amer., August 1968.

14. D. Reddy, and P. Vicens, "Procedure for Segmentation of Connected Speech", J. Audio Eng. Soc., October 1968.

15. D. Reddy, "Consonantal Clustering and Connected Speech Recognition", Proc. Sixth International Congress on Acoustics, Vol. 2, pp. C-57 to C-60, Tokyo, 1968.

16. A. Silvestri and J. Goodman, "Digital Reconstruction of Holographic Images". 1968, NEREM Record, IEEE, Vol. 10, pp. 118-119, 1968.

17. L. Tesler, H. Enea, and K. Colby, "A Directed Graph Representation for Computer Simulation of Belief Systems", Math. Bio. 2, 1968.

1969

1. J. Beauchamp (with H. Von Foerster) (eds.), "Music by Computers", John Wiley, New York, 1969.

2. J. Becker, "The Modeling of Simple Analogic and Inductive Processes in a Semantic Memory System", Proc. International Conf. on Artificial Intelligence, Washington, D.C., 1969.

3. B. Buchanan and G. Sutherland, "Heuristic Dendral: A Program for Generating Hypotheses in Organic Chemistry", in D. Michie (ed.), Machine Intelligence 4, American Elsevier, New York, 1969.

4. B. Buchanan (with C. Churchman), "On the Design of Inductive Systems: Some Philosophical Problems", British Journal for the Philosophy of Science, 20, 1969, pp. 311-323.

5. K. Colby, L. Tesler, and H. Enea, "Experiments with a Search Algorithm for the Data Base of a Human Belief System", Proc. International Conference on Artificial Intelligence, Washington, D.C., 1969.

6. K. Colby and D. Smith, "Dialogues between Humans and Artificial Belief Systems", Proc. International Conference on Artificial Intelligence, Washington, D.C., 1969.

7. A. Duffield, A. Robertson, C. Djerassi, B. Buchanan, G. Sutherland, E. Feigenbaum, and J. Lederberg, "Application of Artificial Intelligence for Chemical Interference II. Interpretation of Low Resolution Mass Spectra of Ketones", J. Amer. Chem. Soc., 91:11, May 1969.

8. J. Feldman, G. Feldman, G. Falk, G. Grape, J. Pearlman, I. Sobel, and J. Tenenbaum, "The Stanford Hand-Eye Project", Proc. International Conf. on Artificial Intelligence, Washington, D.C., 1969.

9. J. Feldman (with P. Rovner), "An Algol-based Associative Language", Comm. ACM, August 1969.

10. T. Ito, "Note on a Class of Statistical Recognition Functions", IEEE Trans. Computers, January 1969.

11. D. Kaplan, "Regular Expressions and the Completeness of Programs", J. Comp. & System Sci., Vol. 3, No. 4, 1969.

12. J. Lederberg, "Topology of Organic Molecules", National Academy of Science, The Mathematical Sciences: a Collection of Essays, MIT Press, Cambridge, 1969.

13. J. Lederberg, G. Sutherland, B. Buchanan, E. Feigenbaum, A. Robertson, A. Duffield, and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference I. The Number of Possible Organic Compounds: Acyclic Structures Containing C, H, O, and N", J. Amer. Chem. Soc., 91:11, May 1969.

14. Z. Manna, "Properties of Programs and the First Order Predicate Calculus", J. ACM, April 1969.

15. Z. Manna, "The Correctness of Programs", J. System and Computer Sciences, May 1969.

16. Z. Manna and A. Pnueli, "Formalization of Properties of Recursively Defined Functions", Proc. ACM Symposium on Computing Theory, May 1969.

17. J. McCarthy and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", in D. Michie (ed.), Machine Intelligence 4, American Elsevier, New York, 1969.

18. U. Montanari, "Continuous Skeletons from Digitized Images", JACM, October 1969.

19. R. Paul, G. Falk, J. Feldman, "The Computer Representation of Simply Described Scenes", Proc. 2ND Illinois Graphics Conference, Univ. Illinois, April 1969.

20. R. Schank and L. Tesler, "A Conceptual Parser for Natural Language", Proc. International Joint Conference on Artificial Intelligence, Washington, D.C., 1969.

21. G. Schroll, A. Duffield, C. Djerassi, B. Buchanan, G. Sutherland, E. Feigenbaum, and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference III. Aliphatic Ethers Diagnosed by Their Low Resolution Mass Spectra and NMR Data", J. Amer. Chem. Soc., 91:26, December 1969.

1970

1. J. Allen and D. Luckham, "An Interactive Theorem-Proving Program" in B. Meltzer and D. Michie (eds.), Machine Intelligence 5, Edinburgh University Press, 1970.

2. B. Buchanan, G. Sutherland, and E. Feigenbaum, "Rediscovering some Problems of Artificial Intelligence in the Context of Organic Chemistry" in B. Meltzer and D. Michie (eds), Machine Intelligence 5, Edinburgh University Press, 1970.

3. B. Buchanan and T. Headrick, "Some Speculation about Artificial Intelligenc and Legal Reasoning", Stanford Law Review, November 1970.

4. B. Buchanan, A. Duffield, A. Robertson, "An Application of Artificial Intelligence to the Interpretation of Mass Spectra", in Mass Spectrometry (G.W. Milne, ed.), Wiley, 1970.

5. A. Buchs, A. Duffield, G. Schroll, C. Djerassi, A. Delfino, B. Buchanan, G. Sutherland, E. Feigenbaum, and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference IV. Saturated Amines Diagnosed by their Low Resolution Mass Spectra and Nuclear Magnetic Resonance Spectra", J. Amer. Chem. Soc. 92:23, November 1970.

6. A. Buchs, A. Delfino, A. Duffield, C. Djerassi, B. Buchanan, E. Feigenbaum, J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference VI. Approach to a General Method of Interpreting Low Resolution Mass Spectra with a Computer", Helvetica Chemica Acta, 53:6, 1970.

7. K. Colby, "Mind and Brain Again", W. McCullough Memorial Vol. of Math. Bio., 1970.

8. E. Feigenbaum, chapter in Readiness to Remember, D. P. Kimball (ed.), Gordon and Breach, 1970.

9. K. Pingle, "Visual Perception by a Computer", in Automatic Interpretation and Classification of Images, Academic Press, New York, 1970.

10. J. Lederberg, G. Sutherland, B. Buchanan, E. Feigenbaum, "A Heuristic Program for Solving a Scientific Inference Problem: Summary of Motivation and Implementation", in M. Mesarovic (ed.), Theoretical Approaches to Non-numerical Problem Solving, Springer-Verlag, New York, 1970.

11. D. Luckham, "Refinement Theorems in Resolution Theory", Proc. 1968 IRIA Symposium in Automatic Deduction, Versailles, France, Springer-Verlag, 1970.

12. D. Luckham (with D. Park and M. Paterson), "On Formalised Computer Programs", J. Comp. & System Sci., Vol. 4, No. 3, June 1970.

13. Z. Manna and J. McCarthy "Properties of Programs and Partial Function Logic" in B. Meltzer and D. Michie (eds.), Machine Intelligence 5, Edinburgh University Press, 1970.

14. Z. Manna, "The Correctness of Non-Deterministic Programs", Artificial Intelligence Journal, Vol. 1, No. 1, 1970.

15. Z. Manna, "Second-order Mathematical Theory of Computation", Proc. ACM Symposium on Theory of Computing, May 1970.

16. Z. Manna and A. Pnueli, "Formalization of Properties of Recursively Defined Functions", J. ACM, July 1970.

17. U. Montanari, "A Note on Minimal Length Polygonal Approximation to a Digitized Contour", Comm. ACM, January 1970.

18. U. Montanari, "On Limit Properties in Digitization Schemes", JACM, April 1970.

19. M. Somalvico (with G. Bracchi), "An Interactive Software System for Computer-aided Design: An Application to Circuit Project", Comm. ACM, September 1970.

20. D. Waterman, "Generalization Learning Techniques for Automating the Learning of Heuristics", J. Artificial Intelligence, Vol. 1, No. 1/2.

1971

1. E. Ashcroft, "Formalization of Properties of Parallel Programs", Machine Intelligence 6, Edinburgh Univ. Press, 1971.

2. E. Feigenbaum, B. Buchanan, J. Lederberg, "On Generality and Problem Solving: A Case Study using the DENDRAL Program", Machine Intelligence 6, Edinburgh Univ. Press, 1971.

3. Y. Sheikh, A. Buchs, A Delfino, B. Buchanan, G. Sutherland, J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference V. An Approach to the Computer Generation of Cyclic Structures. Differentiation Between All the Possible Isometric Ketones of Composition C6H10O", Organic Mass Spectrometry (in Press).

BLANK PAGE

APPENDIX B

THESES

Theses that have been published by the Stanford Artificial Intelligence Project are listed here. Several earned degrees at institutions other than Stanford, as noted. Abstracts of recent A. I. Memos are given in Appendix D.

AIM-43, D. Raj Reddy, AN APPROACH TO COMPUTER SPEECH RECOGNITION BY DIRECT ANALYSIS OF THE SPEECH WAVE, Ph.D. Thesis in Computer Science, September 1966.

AIM-46, S. Persson, SOME SEQUENCE EXTRAPOLATING PROGRAMS: A STUDY OF REPRESENTATION AND MODELING IN INQUIRING SYSTEMS, Ph.D. Thesis in Computer Science, University of California, Berkeley, September 1966.

AIM-47, Bruce Buchanan, LOGICS OF SCIENTIFIC DISCOVERY, Ph.D. Thesis in Philosophy, University of California, Berkeley, December 1966.

AIM-44, James Painter, SEMANTIC CORRECTNESS OF A COMPILER FOR AN ALGOL-LIKE LANGUAGE, Ph.D. Thesis in Computer Science, March 1967.

AIM-56, William Wichman, USE OF OPTICAL FEEDBACK IN THE COMPUTER CONTROL OF AN ARM, Eng. Thesis in Electrical Engineering, August 1967.

AIM-58, M. Callero, AN ADAPTIVE COMMAND AND CONTROL SYSTEM UTILIZING HEURISTIC LEARNING PROCESSES, Ph.D. Thesis in Operations Research, December 1967.

AIM-60, Donald Kaplan, THE FORMAL THEORETIC ANALYSIS OF STRONG EQUIVALENCE FOR ELEMENTAL PROPERTIES, Ph.D. Thesis in Computer Science, July 1968.

AIM-65, Barbara Huberman, A PROGRAM TO PLAY CHESS END GAMES, Ph.D. Thesis in Computer Science, August 1968.

AIM-73, Donald Pieper, THE KINEMATICS OF MANIPULATORS UNDER COMPUTER CONTROL, Ph.D. Thesis in Mechanical Engineering, October 1968.

AIM-74, Donald Waterman, MACHINE LEARNING OF HEURISTICS, Ph.D. Thesis in Computer Science, December 1968.

AIM-83, Roger Schank, A CONCEPTUAL DEPENDENCY REPRESENTATION FOR A COMPUTER ORIENTED SEMANTICS, Ph.D. Thesis in Linguistics, University of Texas, March 1969.

AIM-85, Pierre Vicens, ASPECTS OF SPEECH RECOGNITION BY COMPUTER, Ph.D. Thesis in Computer Science, March 1969.

AIM-92, Victor D. Scheinman, DESIGN OF COMPUTER CONTROLLED MANIPULATOR, Eng. Thesis in Mechanical Engineering, June 1969.

AIM-96, Claude Cordell Green, THE APPLICATION OF THEOREM PROVING TO QUESTION-ANSWERING SYSTEMS, Ph.D. Thesis in Electrical Engineering, August 1969.

AIM-98, James J. Horning, A STUDY OF GRAMMATICAL INFERENCE, Ph.D. Thesis in Computer Science, August 1969.

AIM-106, Michael E. Kahn, THE NEAR-MINIMUM-TIME CONTROL OF OPEN-LOOP ARTICULATED KINEMATIC CHAINS, Ph.D. Thesis in Mechanical Engineering, December 1969.

AIM-121, Irwin Sobel, CAMERA MODELS AND MACHINE PERCEPTION, Ph.D. Thesis in Electrical Engineering, May 1970.

AIM-130, Michael D. Kelly, VISUAL IDENTIFICATION OF PEOPLE BY COMPUTER, Ph.D. Thesis in Computer Science, July 1970.

AIM-132, Gilbert Falk, COMPUTER INTERPRETATION OF IMPERFECT LINE DATA AS A THREE-DIMENSIONAL SCENE, Ph.D. Thesis in Electrical Engineering, August 1970.

AIM-134, Jay Martin Tenenbaum, ACCOMMODATION IN COMPUTER VISION, Ph.D. Thesis in Electrical Engineering, September 1970.

Appendix C

FILM REPORTS

Prints of the following films are available for short-term loan to interested groups without charge. They may be shown only to groups that have paid no admission fee. To make a reservation, write to:

Artificial Intelligence Project Secretary
Computer Science Department
Stanford University
Stanford, California 94305

Alternatively, prints may be purchased at cost (typically $30 to $50) from:

Cine-Chrome Laboratories
4075 Transport St.
Palo Alto, California

1. Art Eisenson and Gary Feldman, "Ellis D. Kroptechev and Zeus, his Marvelous Time-Sharing System", 16mm black and white with sound, 15 minutes, March 1967.

The advantages of time-sharing over standard batch processing are revealed through the good offices of the Zeus time-sharing system on a PDP-1 computer. Our hero, Ellis, is saved from a fate worse than death. Recommended for mature audiences only.

2. Gary Feldman, "Butterfinger", 16mm color with sound, 8 minutes, March 1968.

Describes the state of the hand-eye system at the Artificial Intelligence Project in the fall of 1967. The PDP-6 computer getting visual information from a television camera and controlling an electrical-mechanical arm solves simple tasks involving stacking blocks. The techniques of recognizing the blocks and their positions as well as controlling the arm are briefly presented. Rated "G".

3. Raj Reddy, Dave Espar and Art Eisenson, "Hear Here", 16mm color with sound, 15 minutes, March 1969.

Describes the state of the speech recognition project as of Spring, 1969. A discussion of the problems of speech recognition is followed by two real time demonstrations of the current system. The first shows the computer learning to recognize phrases and second shows how the hand-eye system may be controlled by voice commands. Commands as complicated as "Pick up the small block in the lower lefthand corner", are recognized and the tasks are carried out by the computer controlled arm.

4. Gary Feldman and Donald Peiper, "Avoid", 16mm silent, color, 5 minutes, March 1969.

Reports on a computer program written by D. Peiper for his Ph.D. Thesis. The problem is to move the computer controlled electrical-mechanical arm through a space filled with one or more known obstacles. The program uses heuristics for finding a safe path; the film demonstrates the arm as it moves through various cluttered environments with fairly good success.

Appendix D

ARTIFICIAL INTELLIGENCE MEMOS

These memos report research results. Abstracts of mamos published in 1970 and later are listed here. For an earlier list going back to 1963, see AIM-117.

Interested researchers may obtain available copies upon request to:

Artificial Intelligence Project Secretary
Computer Science Department
Stanford University
Stanford, California 94305

Alternatively, they are available from:

Clearinghouse for Federal Scientific
and Technical Information
Springfield, Virginia 22151

The Clearinghouse charges $3.00 per full size copy and $.95 for a microfiche copy.

1970

AIM-108, Michael D. Kelly, EDGE DETECTION IN PICTURES BY COMPUTER USING PLANNING, January 1970, 28 pages

This paper describes a program for extracting an accurate outline of a man's head from a digital picture. The program accepts as input digital, grey scale pictures containing people standing in front of various backgrounds. The output of the program is an ordered list of the points which form the outline of the head. The edges of background objects and the interior details of the head have been suppressed.

The program is successful because of an improved method for edge detection which uses heuristic planning, a technique drawn from artificial intelligence research in problem solving. A brief, edge detection using planning consists of three steps. A new digital picture is prepared from the original; the new picture is smaller and has less detail. Edges of objects are located in the reduced picture. The edges found in the reduced picture are used as a plan for finding edges in the original picture.

AIM-109, Roger C. Schank, Lawrence Tesler, and Sylvia Weber, SPINOZA II: CONCEPTUAL CASE-BASED NATURAL LANGUAGE ANALYSIS, January 1970, 107 pages.

This paper presents the theoretical changes that have developed in Conceptual Dependency Theory and their ramifications in computer analysis of natural language. The major items of concern are: the elimination of reliance on "grammar rules" for parsing with the emphasis given to conceptual rule based parsing, the development of a conceptual case system to account for the power of

conceptualizations; the categorization of ACT's based on permissible conceptual cases and other criteria. These items are developed and discussed in the context of a more powerful conceptual parser and a theory of language understanding.

AIM-110, Edward Ashcroft and Zohar Manna, FORMALIZATION OF PROPERTIES OF PARALLEL PROGRAMS, February 1970, 58 pages.

In this paper we describe a class of parallel programs and give a formalization of certain properties of such programs in predicate calculus.

Although our programs are syntactically simple, they do exhibit interaction between asynchronous parallel processes, which is the essential feature we wish to consider. The formalization can easily be extended to more complicated programs.

Also presented is a method of simplifying parallel programs, i.e., constructing simpler equivalent programs, based on the "independence" of statements in them. With these simplications our formalization gives a practical method for proving properties of such programs.

AIM-111, Zohar Manna, SECOND-ORDER MATHEMATICAL THEORY OF COMPUTATION, March 1970, 25 pages.

In this work we show that it is possible to formalize all properties regularly observed in (deterministic and non- deterministic) algorithms in second-order predicate calculus.

Moreover, we show that for any given algorithm it suffices to know how to formalize its "partial correctness" by a second-order formula in order to formalize all other properties by second-order formulas.

This result is of special interest since "partial correctness" has already been formalized in second-order predicate calculus for many classes of algorithms.

This paper will be presented at the ACM Symposium on Theory of Computing (May 1970).

AIM-112, Franklin D. Hilf, Kenneth Mark Colby, David C. Smith, and William K. Wittner, MACHINE-MEDIATED INTERVIEWING, March 1970, 27 pages.

A technique of psychiatric interviewing is described in which patient and interviewer communicate by means of remotely located teletypes. Advantages of non-nonverbal communication in the study of the psychiatric interview and in the development of a computer program designed to conduct psychiatric interviews are discussed. Transcripts from representative interviews are reproduced.

AIM-113, Kenneth M. Colby, Franklin R. Hilf, William A. Hall, A MUTE PATIENT'S EXPERIENCE WITH MACHINE-MEDIATED INTERVIEWING, March 1970, 19 pages.

A hospitalized mute patient participated in seven machine-mediated interviews, excerpts of which are presented. After the fifth interview he began to use spoken language for communication. This novel technique is suggested for patients who are unable to participate in the usual vis-a-vis interview.

AIM-114, A.W. Biermann and J.A. Feldman, ON THE SYNTHESIS OF FINITE-STATE ACCEPTORS, April 1970, 31 pages.

Two algorithms are presented for solving the following problem: Given a finite-set S of strings of symbols, find a finite-state machine which will accept the strings of S and possibly some additional strings which "resemble" those of S. The approach used is to directly construct the states and transitions of the acceptor machine from the string information. The algorithms include a parameter which enable one to increase the exactness of the resulting machine's behavior as much as desired by increasing the number of states in the machine. The properties of the algorithms are presented and illustrated with a number of examples.

The paper gives a method for identifying a finite-state language from a randomly chosen finite subset of the language if the subset is large enough and if a bound is known on the number of states required to recognize the language. Finally, we discuss some of the uses of the algorithms and their relationship to the problem of grammatical inference.

AIM-115, Ugo Montanari, ON THE OPTIMAL DETECTION OF CURVES IN NOISY PICTURES, March 1970, 35 pages.

A technique for recognizing systems of lines is presented, in which the heuristic of the problem is not embedded in the recognition algorithm but is expressed in a figure of merit. A multistage decision process is then able to recognize in the input picture the optimal system of lines according to the given figure of merit. Due to the global approach, greater flexibility and adequacy in the particular problem is achieved. The relation between the structure of the figure of merit and the complexity of the optimization process is then discussed. The method described is suitable for parallel processing because the operations relative to each state can be computed in parallel, and the number of stages is equal to the length N of the curves (or to log2(N) if an approximate method is used).

AIM-116, Kenneth Mark Colby, M.D., MIND AND BRAIN, AGAIN, March 1970, 10 pages.

Classical mind-brain questions appear deviant through the lens of an analogy comparing mental processes with computational processes. Problems of reducibility and personal consciousness are also considered in the light of this analogy.

AIM-117, John McCarthy and the Artificial Intelligence Project Staff, E. Feigenbaum, J. Lederberg and the Heuristic DENDRAL Project Staff, PROJECT TECHNICAL REPORT, April 1970, 75 pages.

Current research is reviewed in artificial intelligence and related areas, including representation theory, mathematical theory of computation, models of cognitive processes, speech recognition, and computer vision.

AIM-118, Ugo Montanari, HEURISTICALLY GUIDED SEARCH AND CHROMOSOME MATCHING, April 1970, 29 pages.

Heuristically guided search is a technique which takes systematically into account information from the problem domain for directing the search. The problem is to find the shortest path in a weighted graph from a start vertex Va to a goal vertex Vz: for every intermediate vertex, an estimate is available of the distance to Vz. If this estimate satisfies a consistency assumption, an algorithm by Hart, Nilsson and Raphael is guaranteed to find the optimum, looking at the a priori minimum number of vertices. In this paper, a version of the above algorithm is presented, which is guaranteed to succeed with the minimum amount of storage. An application of this technique to the chromosome matching problem is then shown. Matching is the last stage of automatic chromosome analysis procedures, and can also solve ambiguities in the classification stage. Some peculiarities of this kind of data suggest the use of an heuristically guided search algorithm instead of the standard Edmonds' algorithm. The method that we obtain in this way is proved to exploit the clustering of chromosome data: a linear-quadratic dependence from the number of chromosomes is obtained for perfectly clustered data. Finally, some experimental results are given.

AIM-119, J. Becker, AN INFORMATION-PROCESSING MODEL OF INTERMEDIATE-LEVEL COGNITION, May 1970, 123 pages.

There is a large class of cognitive operations in which an organism adapts its previous experience in order to respond properly to a new situation - for example: the perceptual recognition of objects and events, the prediction of the immediate future (e.g. in tracking a moving object), and the employment of sensory-motor "skills". Taken all together, these highly efficient processes form a cognitive subsystem which is intermediate between the low-level sensory-motor operations and the more deliberate processes of high-level "thought".

The present report describes a formal information-processing model of this "Intermediate-Level" cognitive system. The model includes memory structures for the storage of experience, and processes for responding to new events on the basis of previous experience. In addition, the proposed system contains a large number of mechanisms for making the response-selection process highly efficient, in spite of the vast amount of stored information that the system must cope with. These devices include procedures for heuristically evaluating alternative subprocesses, for guiding the search through memory, and for reorganizing the information in memory into more efficient representations.

AIM-120, K. M. Colby, D.C. Smith, COMPUTER AS CATALYST IN THE TREATMENT OF NONSPEAKING AUTISTIC CHILDREN, April 1970, 32 pages.

Continued experience with a computer-aided treatment method for nonspeaking autistic children has demonstrated improvement effects on thirteen out of a series of seventeen cases. Justification for this conclusion is discussed in detail. Adoption of this method by other research groups is needed for the future development of computer-aided treatment.

AIM-121, Irwin Sobel, CAMERA MODELS AND MACHINE PERCEPTION, May 1970, 89 pages.

We have developed a parametric model for a computer-controlled moveable camera on a pan-tilt head. The model expresses the transform relating object space to image space as a function of the control variables of the camera. We constructed a calibration system for measuring the model parameters which has a demonstrated accuracy more than adequate for our present needs. We have also identified the major source of error in model measurement to be undesired image motion and have developed means of measuring and compensating for some of it and eliminating other parts of it. The system can measure systematic image distortions if they become the major accuracy limitation. We have shown how to generalize the model to handle small systematic errors due to aspects of pan-tilt head geometry not presently accounted for.

We have demonstrated the model's application in stereo vision and have shown how it can be applied as a predictive device in locating objects of interest and centering them in an image.

AIM-122, Roger C. Schank, "SEMANTICS" IN CONCEPTUAL ANALYSIS, May 1970, 56 pages.

This paper examines the question of what a semantic theory should account for. Some aspects of the work of Katz, Fillmore, Lakoff and Chomsky are discussed. "Semantics" is concluded to be the representation problem with respect to conceptual analysis. The beginnings of a solution to this problem are presented in the light

of developments in conceptual dependency theory.

AIM-123, Bruce G. Buchanan, Thomas E. Headrick, SOME SPECULATION ABOUT ARTIFICIAL INTELLIGENCE AND LEGAL REASONING, May 1970, 54 pages.

Legal reasoning is viewed here as a complex problem-solving task to which the techniques of artificial intelligence programming may be applied. Some existing programs are discussed which successfully attack various aspects of the problem, in this and other task domains. It remains an open question, to be answered by intensive research, whether computers can be programmed to do creative legal reasoning. Regardless of the answer, it is argued that much will be gained by the research.

AIM-124, M.M. Astrahan, SPEECH ANALYSIS BY CLUSTERING, OR THE HYPERPHONEME METHOD, June 1970, 22 pages.

In this work, measured speech waveform data was used as a basis for partitioning an utterance into segments and for classifying those segments. Mathematical classifications were used instead of the traditional phonemes or linguistic categories. This involved clustering methods applied to hyperspace points representing periodic samples of speech waveforms. The cluster centers, or hyperphonemes (HPs), were used to classify the sample points by the nearest-neighbor technique. Speech segments were formed by grouping adjacent points with the same classification. A dictionary of 54 different words from a single speaker was processed by this method. 216 utterances, representing four more repetitions by the same speaker each of the original 54 words, were similarly analyzed into strings of hyperphonemes and matched against the dictionary by heuristically developed formulas. 87% were correctly recognized, although almost no attempt was made to modify and improve the initial methods and parameters.

AIM-125, Kenneth M. Colby, Sylvia Weber, and Franklin Hilf, ARTIFICIAL PARANOIA, July 1970, 35 pages.

A case of artificial paranoia has been synthesized in the form of a computer model. Using the test operations of a teletyped psychiatric interview, clinicians judge the input-output behavior of the model to be paranoid. Formal validation of the model will require experiments involving indistinguishability tests.

AIM-126, Donald E. Knuth, EXAMPLES OF FORMAL SEMANTICS, July 1970, 34 pages.

A technique of formal definition, based on relations between "attributes" associated with nonterminal symbols in a context-free grammar, is illustrated by several applications to simple yet typical problems. First we define the basic properties of lambda expressions, involving substitution and renaming of bound variables. Then a simple

programming language is defined using several different points of view. The emphasis is on "declarative" rather than "imperative" forms of definition.

AIM-127, Zohar Manna and Richard J. Waldinger, TOWARDS AUTOMATIC PROGRAM SYNTHESIS, July 1970, 54 pages.

An elementary outline of the theorem-proving approach to automatic program synthesis is given, without dwelling on technical details. The method is illustrated by the automatic construction of both recursive and iterative programs operating on natural numbers, lists, and trees.

In order to construct a program satisfying certain specifications, a theorem induced by those specifications is proved, and the desired program is extracted from the proof. The same technique is applied to transform recursively defined functions into iterative programs, frequently with a major gain in efficiency.

It is emphasized that in order to construct a program with loops or with recursion, the principle of mathematical induction must be applied. The relation between the version of the induction rule used and the form of the program constructed is explored in some detail.

AIM-128, Erik J.Sandewall, REPRESENTING NATURAL-LANGUAGE INFORMATION IN PREDICATE CALCULUS, July 1970, 27 pages.

A set of general conventions are proposed for representing natural language information in many-sorted first order predicate calculus. The purpose is to provide a testing-ground for existing theorem-proving programs.

AIM-129, Shigeru Igarashi, SEMANTICS OF ALGOL-LIKE STATEMENTS, June 1970, 95 pages.

The semantics of elementary Algol-like statements is discussed, mainly based on an axiomatic method.

Firstly, a class of Algol-like statements is introduced by generalized inductive definition, and the interpretation of the statements belonging to it is defined in the form of a function over this class, using the induction principle induced by the above definition. Then a category of program is introduced in order to clarify the concept of equivalence of statements, which becomes a special case of isomorphism in that category.

A revised formal system representing the concept of equivalence of Algol-like statements is presented, followed by elementary metatheorems.

Finally, a process of decomposition of Algol-like statements, which can be regarded as a conceptual compiler, or a constructive description of semantics based on primitive actions, is defined and its correctness is proved formally, by the help of the induced induction principle.

AIM-130, Michael D. Kelly, VISUAL IDENTIFICATION OF PEOPLE BY COMPUTER, July 1970, 238 pages.

This thesis describes a computer program which performs a complex picture processing task. The task is to choose, from a collection of pictures of people taken by a TV camera, those pictures that depict the same person. The primary purpose of this research has been directed toward the development of new techniques for picture processing.

In brief, the program works by finding the location of features such as eyes, nose, or shoulders in the pictures. Individuals are classified by measurements between such features. The interesting and difficult part of the work reported in this thesis is the detection of those features in digital pictures. The nearest neighbor method is used for identification of individuals once a set of measurements has been obtained.

The success of the program is due to and illustrates the heuristic use of context and structure. A new, widely useful, technique called planning has been applied to picture processing. Planning is a term which is drawn from artificial intelligence research in problem solving.

The principal positive result of this research is the use of goal-directed techniques to successfully locate features in cluttered digital pictures. This success has been verified by displaying the results of the feature finding algorithms and comparing these locations with the locations obtained by hand from digital printouts of the pictures. Successful performance in the task of identification of people provides further verification for the feature finding algorithms.

AIM-131, Edward A. Feigenbaum, Bruce G. Buchanan, Joshua Lederberg, ON GENERALITY AND PROBLEM SOLVING: A CASE STUDY USING THE DENDRAL PROGRAM, August 1970, 48 pages.

Heuristic DENDRAL is a computer program written to solve problems of inductive inference in organic chemistry. This paper will use the design of Heuristic DENDRAL and its performance on different problems for a discussion of the following topics:

1. the design for generality;
2. the performance problems attendent upon too much generality
3. the coupling of expertise to the general problem solving

processes,

4. the symbiotic relationship between generality and expertnness of problem solving systems.

We conclude the paper with a view of the design for a general problem solver that is a variant of the "big switch" theory of generality.

AIM-132, Gilbert Falk, COMPUTER INTERPRETATION OF IMPERFECT LINE DATA AS A THREE-DIMENSIONAL SCENE, August 1970, 187 pages.

The major portion of this paper describes a heuristic scene description program. This program accepts as input a scene represented as a line drawing. Based on a set of known object models the program attempts to determine the identity and location of each object viewed. The most significant feature of the program is its ability to deal with imperfect input data.

We also present some preliminary results concerning constraints in projections of planar-faced solids. We show that for a restricted class of projects, 4 points located in 3-space in addition to complete monocular information are sufficient to specify all the visible point locations precisely.

AIM-133, Anthony C. Hearn, REDUCE 2, October 1970, pages.

This manual provides the user with a description of the algebraic programming system REDUCE 2. The capabilities of this system include:

1) Expansion and ordering of rational functions of polynomials, 2) symbolic differentiation of rational functions of polynomials and general functions, 3) substitutions and pattern matching in a wide variety of forms, 4) calculation of the greatest common divisor of two polynomials, 5) automatic and user controlled simplification of expressions, 6) calculations with symbolic matrices, 7) a complete language for symbolic calculations, in which the REDUCE program itself is written, 8) calculations of interest to high energy physicists including spin 1/2 and spin 1 algebra, 9) tensor operations.

AIM-134, Jay Martin Tenenbaum, ACCOMMODATION IN COMPUTER VISION, September 1970, 452 pages.

We describe an evolving computer vision system in which the parameters of the camera are controlled by the computer. It is distinguished from conventional picture processing systems by the fact that sensor accommodation is automatic and treated as an integral part of the recognition process.

A machine, like a person, comes in contact with far more visual information than it can process. Furthermore, no physical sensor can simultaneously provide information about the full range of the environment. Consequently, both man and machine must accommodate

their sensors to emphasize selected characteristics of the environment.

Accommodation improves the reliability and efficiency of machine perception by matching the information provided by the sensor with that required by specific perceptual functions. The advantages of accommodation are demonstrated in the context of five key functions in computer vision: acquisition, contour following, verifying the presence of an expected edge, range-finding, and color recognition.

We have modeled the interaction of camera parameters with scene characteristics to determine the composition of an image. Using a priori knowledge of the environment, the camera is tuned to satisfy the information requirements of a particular task.

Task performance depends implicitly on the appropriateness of available information. If a function fails to perform as expected, and if this failure is attributable to a specific image deficiency, then the relevant accommodation parameters can be refined.

This schema for automating sensor accommodation can be applied in a variety of perceptual domains.

AIM-135, David Canfield Smith, MLISP, October 1970, 99 pages.

MLISP is a high level list-processing and symbol-manipulation language based on the programming language LISP. MLISP programs are translated into LISP programs and then executed or compiled. MLISP exists for two purposes: (1) to facilitate the writing and understanding of LISP programs; (2) to remedy certain important deficiencies in the list-processing ability of LISP.

AIM-136, George M. White, MACHINE LEARNING THROUGH SIGNATURE TREES, APPLICATION TO HUMAN SPEECH, October 1970, 40 pages.

Signature tree "machine learning", pattern recognition heuristics are investigated for the specific problem of computer recognition of human speech. When the data base of given utterances is insufficient to establish trends with confidence, a large number of feature extractors must be employed and "recognition" of an unknown pattern made by comparing its feature values with those of known patterns. When the data base is replete, a "signature" tree can be constructed and recognition can be achieved by the evaluation of a select few features. Learning results from selecting an optimal minimal set of features to achieve recognition. Properties of signature trees and the heuristics for this type of learning are of primary interest in this exposition.

AIM-137, Donald E. Knuth, AN EMPIRICAL STUDY OF FORTRAN IN USE, November 1970, 44 pages.

A sample of programs, written in Fortran by a wide variety of people for a wide variety of applications, was chosen "at random" in an attempt to discover quantitatively "what programmers really do". Statistical results of this survey are presented here, together with some of their apparent implications for future work in compiler design. The principle conclusion which may be drawn is the importance of a program "profile", namely a table of frequency counts which record how often each statement is performed in a typical run: there are strong indications that profile-keeping should become a standard practice in all computer systems, for casual users as well as system programmers. Some new approaches to compiler optimization are also suggested. This paper is the report of a three month study undertaken by the author and about a dozen students and representatives of the software industry during the summer of 1970.

AIM-138, E. Ashcroft and Z. Manna, THE TRANSLATION OF 'GO-TO' PROGRAMS TO 'WHILE' PROGRAMS, November 1970, 28 pages.

In this paper we show that every flowchart program can be written without 'go-to' statements by using 'while' statements. The main idea is to introduce new variables to preserve the values of certain variables at particular points in the program; or alternatively, to introduce special boolean variables to keep information about the course of the computation. The new programs preserve the 'topology' of the original program, and are of the same order of efficiency. We also show that this cannot be done in general without adding variables.

AIM-139, Zohar Manna, MATHEMATICAL THEORY OF PARTIAL CORRECTNESS, December 1970, 24 pages.

In this work we show that it is possible to express most properties regularly observed in algorithms in terms of 'partial correctness' (i.e., the property that the final results of the algorithm, if any, satisfy some given input-output relation). This result is of special interest since 'partial correctness' has already been formulated in predicate calculus and in partial function logic for many classes of algorithms.

1971

AIM-140, Roger C. Schank, INTENTION, MEMORY, AND COMPUTER UNDERSTANDING, January 1971, 59 pages.

Procedures are described for discovering the intention of a speaker by relating the Conceptual Dependence representation of the speaker's utterance to the computer's world model such that simple implications can be made. These procedures function at levels higher than that of the sentence by allowing for predictions based on context and the structure of the memory. Computer understanding of natural language is shown to consist of the following parts: assigning a conceptual representation to an input; relating that representation to the

memory such as to extract the intention of the speaker; and selecting the correct response type triggered by such an utterance according to the situation.

AIM-141, Bruce G. Buchanan, Edward A. Feigenbaum, and Joshua Lederberg, THE HEURISTIC DENDRAL PROGRAM FOR EXPLAINING EMPIRICAL DATA, February 1971, 20 pages.

The Heuristic DENDRAL program uses an information processing model of scientific reasoning to explain experimental data in organic chemistry. This report summarizes the organization and results of the program for computer scientists. The program is divided into three main parts: planning, structure generation, and evaluation.

The planning phase infers constraints on the search space from the empirical data input to the system. The structure generation phase searches a tree whose termini are models of chemical models using pruning heuristics of various kinds. The evaluation phase tests the candidate structures against the original data. Results of the program's analyses of some test data are discussed.

AIM-142, Robin Milner, AN ALGEBRAIC DEFINITION OF SIMULATION BETWEEN PROGRAMS, February 1971, 21 pages.

A simulation relation between programs is defined which is quasi-ordering. Mutual simulation is then an equivalence relation, and by dividing out by it we abstract from a program such details as how the sequencing is controlled and how data is represented. The equivalence classes are approximations to the algorithms which are realized, or expressed, by their member programs.

A technique is given and illustrated for proving simulation and equivalence of programs; there is an analogy with Floyd's technique for proving correctness of programs. Finally, necessary and sufficient conditions for simulation are given.

Appendix E

OPERATING NOTES

Stanford Artificial Intelligence Laboratory Operating Notes (SAILONs) describe the operation of computer programs and equipment and are intended for project use. This annotated list omits obsolete notes.

The laboratory has a dual-processor (DEC PDP-10/PDP-6) time-shared computer with 131 thousand words of core memory backed by a swapping disk (20 million bits per second transfer rate) and an IBM 2314 disk file. Online terminals include 40 display consoles and 15 Teletype terminals. Other online equipment includes TV cameras, mechanical arms, audio input and output.

SAILON-2.1, W. Weiher, "Calcomp Plot Routines", September 1968.

SAILON-3.1, B. Baumgart, "How to Do It and Summaries of Things", March 1969. An introductory summary of system features (obsolescent).

SAILON-8, S. Russell, "Recent Additions to FORTRAN Library", March 1967.

SAILON-9, P. Petit, "Electronic Clock", March 1967. Electronic clock attached to the system gives time in micro-seconds, seconds, minutes, hours, day, month, and year. You have to remember whether it's B.C. or A.D.

SAILON-11, P. Petit, "A Recent Change to the Stanford PDP-6 Hardware", March 1967. The PDP-6 has been changed so that user programs can do their own I/O to devices numbered 700 and above.

SAILON-21, A. Grayson, "The A-D Converter", June 1967.

SAILON-21 Addendum 1, E. Panofsky, "A/D Converter Multiplexer Patch Panel and Channel Assignments as of 1-9-69", January 1969.

SAILON-24, S. Russell, "PDP-6 I/O Device Number Summary", August 1967.

SAILON-25, S. Russell, "The Miscellaneous Outputs", August 1967. Gives bit assignments for output to hydraulic arm and TV camera positioning.

SAILON-26.2, P. Petit, "FAIL", April 1970. Describes one-pass assembler that is about five times as fast as MACRO and has a more powerful macro processor.

SAILON-28.3, L. Quam, "Stanford LISP 1.6 Manual", September 1969. Describes the LISP interpreter and compiler, the editor ALVINE, and other aspects of this venerated list processing system.

SAILON-29, W. Weiher, "Preliminary Description of the Display Processor", August 1967. III display system from the programmer's viewpoint.

SAILON-31, J. Sauter, "Disc Diagnostic", October 1967. A program to test the Librascope Disk and its interface.

SAILON-35.2, K. Pingle, "Hand-Eye Library File", April 1970.

SAILON-36, G. Feldman, "Fourier Transform Subroutine", June 1968. FORTRAN subroutine performs one-dimensional Fast Fourier Transform.

SAILON-37, S. Russell and L. Earnest, "A.I. Laboratory Users Guide", June 1968. Orientation and administrative procedures.

SAILON-37, Supplement 1, J. McCarthy, "A.I. Laboratory Users Guide", June 1968. Hard-line administration.

SAILON-38, P. Vicens, "New Speech Hardware", August 1968. Preprocessor for input to speech recognition systems.

SAILON-39, J. Sauter and D. Swinehart, "SAVE", August 1968. Program for saving and restoring a single user's disk files on magnetic tape.

SAILON-41, L. Quam, "SMILE at LISP", September 1968. A package of useful LISP functions.

SAILON-42, G. Falk, "Vidicon Noise Measurements", September 1968. Measurements of spatial and temporal noise on Cohu vidicon camera connected to the computer.

SAILON-43, A. Moorer, "DAEMON - Disk Dump and Restore", September 1968. Puts all or selected files on magnetic tape. New version described in SAILON-54.

SAILON-44, A. Moorer, "FCROX - MACROX to Fail Converter", September 1968. Converts MACRO programs to FAIL format, with a few annotated exceptions.

SAILON-45, A. Hearn, "REDUCE Implementation Guide", October 1968. Describes the procedure for assembling REDUCE (a symbolic computation system) in any LISP system.

SAILON-46, W. Weiher, "Loader Input Format", October 1968.

SAILON-47, and 47 Supplement 1, J. Sauter and J. Singer, "Known Programming Differences Between the PDP-6 and PDP-10" November 1968.

SAILON-49, A. Hearn, "Service Routines for Standard LISP Users", February 1969.

SAILON-50,2, S. Savitzky, "Son of Stopgap", April 1970. A line-number-oriented text editor with string search and substitution commands and hyphenless text justification.

SAILON-52,1, A. Moorer, "System Bootstrapper's Manual", February 1969. How to bring back the system from various states of disarray.

SAILON-53, R. Neely and J. Beauchamp, "Some FORTRAN I/O Humanization Techniques", March 1969. How to live with FORTRAN crockery.

SAILON-54,2, A. Moorer, "Stanford A-I Project Monitor Manual: Chapter I - Console Commands", September 1970. How to talk to the timesharing system.

SAILON-55,2, A. Moorer, "Stanford A-I Project monitor Manual: Chapter II - User Programming", September 1970. Machine language commands to the timesharing system.

SAILON-56, T. Panofsky, "Stanford A-I Facility Manual", Computer equipment features (in preparation).

SAILON-57, D. Swinehart and R. Sproull, "SAIL", November 1969. ALGOL-60 compiler with LEAP constructs and string processing.

SAILON-58, P. Petit, "RAID", September 1969. Display-oriented machine language debugging package.

SAILON-59, A. Moorer, "MONMON", October 1969. Lets you peer into the TS monitor.

SAILON-60, L. Earnest, "Documentation Services", February 1970. Text preparation by computer is often cheaper then typewriters. Facilities for text preparation and reproduction are discussed.

SAILON-61, R. Helliwell, "COPY", January 1971. A program for moving files from one place to another, often with interesting side effects.